
OOP

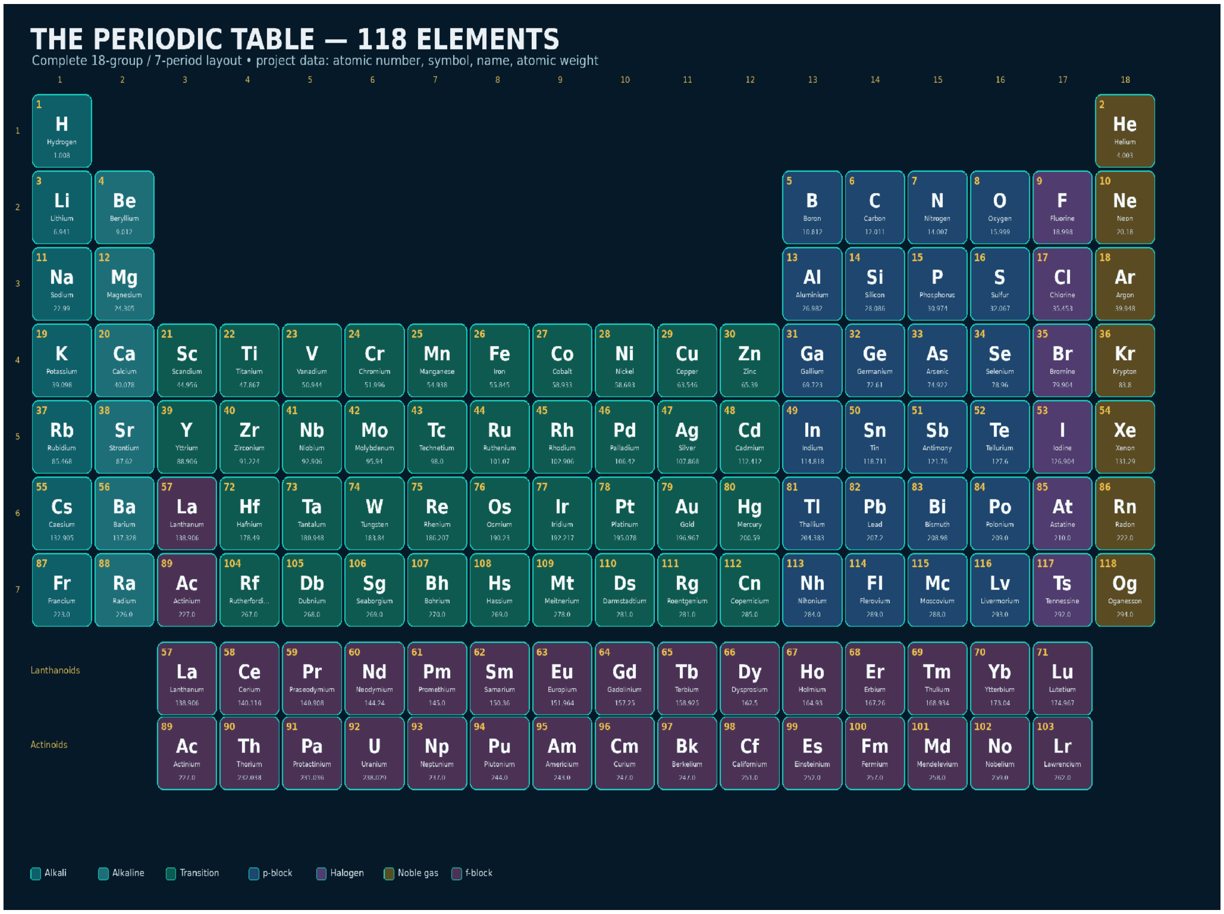
طوبةٌ بـ طوبة — Brick by Brick.

Part III Project
The Periodic Table as an Educational and Engineering Platform for Integrating SQL, Artificial Intelligence, and Quantum Computing.

مشروع الجزء الثالث

الجدول الدوري كمنصة تعليمية وهندسية للتكامل بين SQL والذكاء الاصطناعي والحوسبة الكمومية.

.Email: mohamedelzoghpe905@gmail.com



إلى التحليل والتمثيل SQL هذا الجزء يستخدم الجدول الدوري مثالًا مضبوطًا لتعليم انتقال البيانات من الكمومي.

This part uses the periodic table as a controlled example for moving data from SQL into analysis and quantum representation.

نطاق المشروع وحدوده العلمية

Scientific Scope and Boundaries

الجدول الدوري للعناصر — اللوحة الكاملة للعناصر الـ118.

Periodic Table of the Elements — the complete 118-element project plate.

دليل العناصر داخل المشروع

Element Data Dictionary for the Project.

في الصفحة التالية نربط كل عنصر بسجله العددي في قاعدة البيانات وبالمسارات التحليلية التي يستخدمها المشروع. العدد الذري والرمز والوزن الذري هي حقول أساسية في سجل العنصر؛ وتنتقل البيانات نفسها إلى المصفوفات، و Feature Store، وخطوط التحليل والتصنيف. الهيدروجين هو المثال الذي يُستخدم صراحةً في مسار VQE، بينما تبقى بقية العناصر سجلات وميزات تحليلية ضمن المنصة.

The following pages connect every element to its numerical record in the database and to the analytical paths used by the project. Atomic number, symbol, and atomic weight are core element fields; the same records feed matrices, the Feature Store, and analytical and classification workflows. Hydrogen is the element explicitly used in the VQE workflow, while the remaining elements remain database records and analytical features within the platform.

العدد الذري **Atomic Number**	الرمز **Symbol**	العنصر **Element**	الوزن الذري **Atomic Weight**	التصنيف الدوري **Periodic Classification**	دور العنصر في المشروع **Project Role**
1	H	الهيدروجين Hydrogen	1.008	المجموعة 1 Group 1 الدورة 1 / Period 1 فلز قلوي / Alkali	سجل SQL + مدخل للمصفوفات والتعلّم الآلي + مثال VQE SQL record + matrix and ML input + VQE example
2	He	الهيليوم Helium	4.003	المجموعة 18 Group 18 الدورة 1 / Period 1 غاز نبيل / Noble gas	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
3	Li	الليثيوم Lithium	6.941	المجموعة 1 Group 1 الدورة 2 / Period 2 فلز قلوي / Alkali	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
4	Be	البريليوم Beryllium	9.012	المجموعة 2 Group 2 الدورة 2 / Period 2 Alkaline / فلز قلوي ترابي	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
5	B	البورون Boron	10.812	المجموعة 13 Group 13 الدورة 2 / Period 2 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
6	C	الكربون Carbon	12.011	المجموعة 14 Group 14 الدورة 2 / Period 2 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
7	N	النيتروجين Nitrogen	14.007	المجموعة 15 Group 15 الدورة 2 / Period 2 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
8	O	الأكسجين Oxygen	15.999	المجموعة 16 Group 16 الدورة 2 / Period 2	سجل SQL + مدخل للمصفوفات والتعلّم الآلي

العدد الذري **Atomic Number**	الرمز **Symbol**	العنصر **Element**	الوزن الذري **Atomic Weight**	التصنيف الدوري **Periodic Classification**	دور العنصر في المشروع **Project Role**
				كتلة p / p-block	SQL record + matrix and ML input
9	F	الفلور Fluorine	18.998	المجموعة 17 Group 17 الدورة 2 / Period 2 هالوجين / Halogen	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
10	Ne	النيون Neon	20.18	المجموعة 18 Group 18 الدورة 2 / Period 2 غاز نبيل / Noble gas	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
11	Na	الصوديوم Sodium	22.99	المجموعة 1 Group 1 الدورة 3 / Period 3 فلز قلوي / Alkali	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
12	Mg	المغنيسيوم Magnesium	24.305	المجموعة 2 Group 2 الدورة 3 / Period 3 Alkaline / فلز قلوي ترابي	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
13	Al	الألومنيوم Aluminium	26.982	المجموعة 13 Group 13 الدورة 3 / Period 3 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
14	Si	السيليكون Silicon	28.086	المجموعة 14 Group 14 الدورة 3 / Period 3 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
15	P	الفوسفور Phosphorus	30.974	المجموعة 15 Group 15 الدورة 3 / Period 3 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
16	S	الكبريت Sulfur	32.067	المجموعة 16 Group 16 الدورة 3 / Period 3 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
17	Cl	الكلور Chlorine	35.453	المجموعة 17 Group 17 الدورة 3 / Period 3 هالوجين / Halogen	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
18	Ar	الأرجون Argon	39.948	المجموعة 18 Group 18 الدورة 3 / Period 3 غاز نبيل / Noble gas	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
19	K	البوتاسيوم Potassium	39.098	المجموعة 1 Group 1 الدورة 4 / Period 4 فلز قلوي / Alkali	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
20	Ca	الكالسيوم Calcium	40.078	المجموعة 2 Group 2 الدورة 4 / Period 4 Alkaline / فلز قلوي ترابي	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
21	Sc	السكانديوم Scandium	44.956	المجموعة 3 Group 3 الدورة 4 / Period 4 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
22	Ti	التيتانيوم Titanium	47.867	المجموعة 4 Group 4 الدورة 4 / Period 4 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
23	V	الفاناديوم Vanadium	50.944	المجموعة 5 Group 5 الدورة 4 / Period 4 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input

العدد الذري Atomic Number	الرمز Symbol	العنصر Element	الوزن الذري Atomic Weight	التصنيف الدوري Periodic Classification	دور العنصر في المشروع Project Role
24	Cr	الكروم Chromium	51.996	المجموعة 6 Group 6 الدورة 4 / Period 4 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
25	Mn	المنغنيز Manganese	54.938	المجموعة 7 Group 7 الدورة 4 / Period 4 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
26	Fe	الحديد Iron	55.845	المجموعة 8 Group 8 الدورة 4 / Period 4 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
27	Co	الكوبالت Cobalt	58.933	المجموعة 9 Group 9 الدورة 4 / Period 4 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
28	Ni	النيكل Nickel	58.693	المجموعة 10 Group 10 الدورة 4 / Period 4 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
29	Cu	النحاس Copper	63.546	المجموعة 11 Group 11 الدورة 4 / Period 4 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
30	Zn	الزنك Zinc	65.39	المجموعة 12 Group 12 الدورة 4 / Period 4 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
31	Ga	الجاليوم Gallium	69.723	المجموعة 13 Group 13 الدورة 4 / Period 4 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
32	Ge	الجرمانيوم Germanium	72.61	المجموعة 14 Group 14 الدورة 4 / Period 4 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
33	As	الزرنيخ Arsenic	74.922	المجموعة 15 Group 15 الدورة 4 / Period 4 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
34	Se	السيلينيوم Selenium	78.96	المجموعة 16 Group 16 الدورة 4 / Period 4 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
35	Br	البروم Bromine	79.904	المجموعة 17 Group 17 الدورة 4 / Period 4 هالوجين / Halogen	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
36	Kr	الكريبتون Krypton	83.8	المجموعة 18 Group 18 الدورة 4 / Period 4 غاز نبيل / Noble gas	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
37	Rb	الروبيديوم Rubidium	85.468	المجموعة 1 Group 1 الدورة 5 / Period 5 فلز قلوي / Alkali	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
38	Sr	السترونشيوم Strontium	87.62	المجموعة 2 Group 2 الدورة 5 / Period 5 Alkaline / فلز قلوي ترابي	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
39	Y	الإيتريوم Yttrium	88.906	المجموعة 3 Group 3 الدورة 5 / Period 5	سجل SQL + مدخل للمصفوفات والتعلّم الآلي

العدد الذري Atomic Number	الرمز Symbol	العنصر Element	الوزن الذري Atomic Weight	التصنيف الدوري Periodic Classification	دور العنصر في المشروع Project Role
				فلز انتقالي / Transition	SQL record + matrix and ML input
40	Zr	الزركونيوم Zirconium	91.224	المجموعة 4 Group 4 الدورة 5 / Period 5 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
41	Nb	النيوبيوم Niobium	92.906	المجموعة 5 Group 5 الدورة 5 / Period 5 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
42	Mo	الموليبدينوم Molybdenum	95.94	المجموعة 6 Group 6 الدورة 5 / Period 5 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
43	Tc	التكنيتيوم Technetium	98.0	المجموعة 7 Group 7 الدورة 5 / Period 5 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
44	Ru	الروثينيوم Ruthenium	101.07	المجموعة 8 Group 8 الدورة 5 / Period 5 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
45	Rh	الروديوم Rhodium	102.906	المجموعة 9 Group 9 الدورة 5 / Period 5 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
46	Pd	البلاديوم Palladium	106.42	المجموعة 10 Group 10 الدورة 5 / Period 5 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
47	Ag	الفضة Silver	107.868	المجموعة 11 Group 11 الدورة 5 / Period 5 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
48	Cd	الكادميوم Cadmium	112.412	المجموعة 12 Group 12 الدورة 5 / Period 5 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
49	In	الإنديوم Indium	114.818	المجموعة 13 Group 13 الدورة 5 / Period 5 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
50	Sn	القصدير Tin	118.711	المجموعة 14 Group 14 الدورة 5 / Period 5 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
51	Sb	الأنتيمون Antimony	121.76	المجموعة 15 Group 15 الدورة 5 / Period 5 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
52	Te	التيلوريوم Tellurium	127.6	المجموعة 16 Group 16 الدورة 5 / Period 5 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
53	I	اليود Iodine	126.904	المجموعة 17 Group 17 الدورة 5 / Period 5 هالوجين / Halogen	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
54	Xe	الزينون Xenon	131.29	المجموعة 18 Group 18 الدورة 5 / Period 5 غاز نبيل / Noble gas	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input

العدد الذري Atomic Number	الرمز Symbol	العنصر Element	الوزن الذري Atomic Weight	التصنيف الدوري Periodic Classification	دور العنصر في المشروع Project Role
55	Cs	السيزيوم Caesium	132.905	المجموعة 1 Group 1 الدورة 6 / Period 6 فلز قلوي / Alkali	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
56	Ba	الباريوم Barium	137.328	المجموعة 2 Group 2 الدورة 6 / Period 6 فلز قلوي ترابي / Alkaline	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
57	La	اللانثانوم Lanthanum	138.906	المجموعة 3 Group 3 الدورة 6 / Period 6 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
58	Ce	السيريوم Cerium	140.116	المجموعة 3 Group 3 الدورة 6 / Period 6 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
59	Pr	البراسيوديميوم Praseodymium	140.908	المجموعة 3 Group 3 الدورة 6 / Period 6 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
60	Nd	النيوديميوم Neodymium	144.24	المجموعة 3 Group 3 الدورة 6 / Period 6 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
61	Pm	البروميثيوم Promethium	145.0	المجموعة 3 Group 3 الدورة 6 / Period 6 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
62	Sm	الساماريوم Samarium	150.36	المجموعة 3 Group 3 الدورة 6 / Period 6 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
63	Eu	اليوروبيوم Europium	151.964	المجموعة 3 Group 3 الدورة 6 / Period 6 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
64	Gd	الجادولينيوم Gadolinium	157.25	المجموعة 3 Group 3 الدورة 6 / Period 6 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
65	Tb	التربيوم Terbium	158.925	المجموعة 3 Group 3 الدورة 6 / Period 6 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
66	Dy	الديسبروسيوم Dysprosium	162.5	المجموعة 3 Group 3 الدورة 6 / Period 6 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
67	Ho	الهولميوم Holmium	164.93	المجموعة 3 Group 3 الدورة 6 / Period 6 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
68	Er	الإربيوم Erbium	167.26	المجموعة 3 Group 3 الدورة 6 / Period 6 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
69	Tm	الثوليوم Thulium	168.934	المجموعة 3 Group 3 الدورة 6 / Period 6 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
70	Yb	الإيتربيوم Ytterbium	173.04	المجموعة 3 Group 3 الدورة 6 / Period 6	سجل SQL + مدخل للمصفوفات والتعلّم الآلي

العدد الذري Atomic Number	الرمز Symbol	العنصر Element	الوزن الذري Atomic Weight	التصنيف الدوري Periodic Classification	دور العنصر في المشروع Project Role
				كتلة f / f-block	SQL record + matrix and ML input
71	Lu	اللوتيتيوم Lutetium	174.967	المجموعة 3 Group 3 الدورة 6 / Period 6 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
72	Hf	الهافنيوم Hafnium	178.49	المجموعة 4 Group 4 الدورة 6 / Period 6 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
73	Ta	التنتالوم Tantalum	180.948	المجموعة 5 Group 5 الدورة 6 / Period 6 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
74	W	التنجستن Tungsten	183.84	المجموعة 6 Group 6 الدورة 6 / Period 6 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
75	Re	الرينيوم Rhenium	186.207	المجموعة 7 Group 7 الدورة 6 / Period 6 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
76	Os	الأوزميوم Osmium	190.23	المجموعة 8 Group 8 الدورة 6 / Period 6 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
77	Ir	الإيريديوم Iridium	192.217	المجموعة 9 Group 9 الدورة 6 / Period 6 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
78	Pt	البلاتين Platinum	195.078	المجموعة 10 Group 10 الدورة 6 / Period 6 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
79	Au	الذهب Gold	196.967	المجموعة 11 Group 11 الدورة 6 / Period 6 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
80	Hg	الزئبق Mercury	200.59	المجموعة 12 Group 12 الدورة 6 / Period 6 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
81	Tl	الثاليوم Thallium	204.383	المجموعة 13 Group 13 الدورة 6 / Period 6 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
82	Pb	الرصاص Lead	207.2	المجموعة 14 Group 14 الدورة 6 / Period 6 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
83	Bi	البزموت Bismuth	208.98	المجموعة 15 Group 15 الدورة 6 / Period 6 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
84	Po	البولونيوم Polonium	209.0	المجموعة 16 Group 16 الدورة 6 / Period 6 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
85	At	الأستاتين Astatine	210.0	المجموعة 17 Group 17 الدورة 6 / Period 6 هالوجين / Halogen	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input

العدد الذري **Atomic Number**	الرمز **Symbol**	العنصر **Element**	الوزن الذري **Atomic Weight**	التصنيف الدوري **Periodic Classification**	دور العنصر في المشروع **Project Role**
86	Rn	الرادون Radon	222.0	المجموعة 18 Group 18 الدورة 6 / Period 6 غاز نبيل / Noble gas	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
87	Fr	الفرانسيوم Francium	223.0	المجموعة 1 Group 1 الدورة 7 / Period 7 فلز قلوي / Alkali	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
88	Ra	الراديوم Radium	226.0	المجموعة 2 Group 2 الدورة 7 / Period 7 فلز قلوي ترابي / Alkaline	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
89	Ac	الأكتينيوم Actinium	227.0	المجموعة 3 Group 3 الدورة 7 / Period 7 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
90	Th	الثوريوم Thorium	232.038	المجموعة 3 Group 3 الدورة 7 / Period 7 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
91	Pa	البروتكتينيوم Protactinium	231.036	المجموعة 3 Group 3 الدورة 7 / Period 7 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
92	U	اليورانيوم Uranium	238.029	المجموعة 3 Group 3 الدورة 7 / Period 7 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
93	Np	النبتونيوم Neptunium	237.0	المجموعة 3 Group 3 الدورة 7 / Period 7 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
94	Pu	البلوتونيوم Plutonium	244.0	المجموعة 3 Group 3 الدورة 7 / Period 7 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
95	Am	الأمريسيوم Americium	243.0	المجموعة 3 Group 3 الدورة 7 / Period 7 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
96	Cm	الكوريوم Curium	247.0	المجموعة 3 Group 3 الدورة 7 / Period 7 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
97	Bk	البركيليوم Berkelium	247.0	المجموعة 3 Group 3 الدورة 7 / Period 7 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
98	Cf	الكاليفورنيوم Californium	251.0	المجموعة 3 Group 3 الدورة 7 / Period 7 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
99	Es	الأينشتاينيوم Einsteinium	252.0	المجموعة 3 Group 3 الدورة 7 / Period 7 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
100	Fm	الفرميوم Fermium	257.0	المجموعة 3 Group 3 الدورة 7 / Period 7 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
101	Md	المندليفيوم Mendelevium	258.0	المجموعة 3 Group 3 الدورة 7 / Period 7	سجل SQL + مدخل للمصفوفات والتعلّم الآلي

العدد الذري Atomic Number	الرمز Symbol	العنصر Element	الوزن الذري Atomic Weight	التصنيف الدوري Periodic Classification	دور العنصر في المشروع Project Role
				كتلة f / f-block	SQL record + matrix and ML input
102	No	النوبليوم Nobelium	259.0	المجموعة 3 Group 3 الدورة 7 / Period 7 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
103	Lr	اللورنسيوم Lawrencium	262.0	المجموعة 3 Group 3 الدورة 7 / Period 7 كتلة f / f-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
104	Rf	الرذرفورديوم Rutherfordium	267.0	المجموعة 4 Group 4 الدورة 7 / Period 7 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
105	Db	الدوبنيوم Dubnium	268.0	المجموعة 5 Group 5 الدورة 7 / Period 7 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
106	Sg	السيبورجيوم Seaborgium	269.0	المجموعة 6 Group 6 الدورة 7 / Period 7 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
107	Bh	البوهريوم Bohrium	270.0	المجموعة 7 Group 7 الدورة 7 / Period 7 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
108	Hs	الهاسيوم Hassium	269.0	المجموعة 8 Group 8 الدورة 7 / Period 7 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
109	Mt	الميتنريوم Meitnerium	278.0	المجموعة 9 Group 9 الدورة 7 / Period 7 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
110	Ds	الدارمشتاتيوم Darmstadtium	281.0	المجموعة 10 Group 10 الدورة 7 / Period 7 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
111	Rg	الرونتجينيوم Roentgenium	281.0	المجموعة 11 Group 11 الدورة 7 / Period 7 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
112	Cn	الكوبرنيسيوم Copernicium	285.0	المجموعة 12 Group 12 الدورة 7 / Period 7 فلز انتقالي / Transition	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
113	Nh	النيهونيوم Nihonium	284.0	المجموعة 13 Group 13 الدورة 7 / Period 7 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
114	Fl	الفليروفيوم Flerovium	289.0	المجموعة 14 Group 14 الدورة 7 / Period 7 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
115	Mc	الموسكوفيوم Moscovium	288.0	المجموعة 15 Group 15 الدورة 7 / Period 7 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
116	Lv	الليفرموريوم Livermorium	293.0	المجموعة 16 Group 16 الدورة 7 / Period 7 كتلة p / p-block	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input

العدد الذري Atomic Number	الرمز Symbol	العنصر Element	الوزن الذري Atomic Weight	التصنيف الدوري Periodic Classification	دور العنصر في المشروع Project Role
117	Ts	التينيسين Tennessine	292.0	المجموعة 17 Group 17 الدورة 7 / Period 7 هالوجين / Halogen	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input
118	Og	الأوغانيسون Oganesson	294.0	المجموعة 18 Group 18 الدورة 7 / Period 7 غاز نبيل / Noble gas	سجل SQL + مدخل للمصفوفات والتعلّم الآلي SQL record + matrix and ML input

مقدمة المشروع

Project Introduction

أمامك الآن مشروع تطبيقي يواصل ما تعلمته في "طوبة بطوبة": من تمثيل البيانات الحيوية والبرمجة إلى قواعد البيانات، والتعلم الآلي، والحوسبة الكمومية. الهدف هنا أن ترى كيف تتحول المفاهيم التي تعلمتها في الأجزاء السابقة إلى نظام حسابي يمكن فحصه وتشغيله وتوثيقه.

You now have an applied project that continues the Brick by Brick path from biological data and programming to databases, machine learning, and quantum computing. The goal is to let you see how the concepts developed in the earlier parts become a computational system that can be inspected, executed, and documented.

يعرض لك هذا المشروع الجدول الدوري بوصفه مثالًا عمليًا لبيانات منظمة يمكن تمثيلها في SQL، ونمذجتها بأسلوب البرمجة كائنية التوجه (OOP)، ثم تحويلها إلى ميزات عددية وتغذيتها بمسارات التعلم الآلي والتمثيل الكمومي. هذا المثال لا يمثل جينومًا كاملًا؛ فبيانات جينوم واحد قد تكون أكبر بكثير من أن تُكتب أو تُعرض بالكامل داخل كتاب. الهدف هو أن ترى كيف تساعدك البنية البرمجية وقواعد البيانات والحوسبة المتقدمة على التعامل مع أحجام بيانات علمية متزايدة، مع إبقاء سرعة التحليل مرتبطة بحدود الدقة والتحقق والسلامة الأخلاقية.

This project uses the periodic table as a practical example of structured data that can be represented in SQL, modeled with object-oriented programming (OOP), converted into numerical features, and passed through machine-learning and quantum-representation workflows. This example is not a complete genome: a single genome dataset can be far too large to reproduce in full inside a book. The purpose is to show how software architecture, databases, and advanced computing help you work

with growing scientific datasets while keeping speed tied to accuracy, validation, and ethical safeguards.

لأن بيانات العناصر تجمع بين هويات ثابتة، وعلاقات متكررة، وأنماط دورية، وخواص قابلة للقياس، فهي مناسبة لبناء مثال هندسي متدرج من قاعدة البيانات إلى التحليل العددي ثم التعلم الآلي والحوسبة الكمومية.

Because element data combines stable identities, repeated relationships, periodic patterns, and measurable properties, it provides a useful engineering example that can progress from database design to numerical analysis, machine learning, and quantum computing.

ماذا ستفعل في هذا المشروع؟

What will you build in this project?

1. ستصمم قاعدة بيانات SQL كاملة للجدول الدوري.

1. Design a complete SQL database for the periodic table.

2. ستدخل بيانات 118 عنصرًا بدقة علمية موثقة.

2. Load the 118 elements with documented scientific data.

3. ستستعلم هذه البيانات باستخدام SQL لفهم العلاقات بين العناصر.

3. Query the data with SQL to explore relationships among elements.

4. ستحول البيانات إلى مصفوفات NumPy للتحليل العددي.

4. Convert the data into NumPy matrices for numerical analysis.

5. ستطبق تحليل المكونات الرئيسية (PCA) لاستكشاف البنية الخفية للجدول.

5. Apply PCA to explore latent structure in the feature space.

6. ستستخدم التجميع (Clustering) لتصنيف العناصر آليًا.

6. Use clustering as an unsupervised exploratory method.

7. ستبني مستودع ميزات (Feature Store) للاستخدام في الذكاء الاصطناعي.

7. Build a Feature Store for machine-learning features.

8. ستدرب نماذج تعلم آلي للتنبؤ بخصائص العناصر وتصنيفها.

8. Train machine-learning models for prediction and classification.

9. ستمهد الطريق للحوسبة الكمومية بتمثيل الخواص الذرية كحالات كمومية.

9. Prepare a computational bridge to quantum-state representations.

10. ستحاكي ذرة الهيدروجين باستخدام خوارزمية VQE الكمومية.

10. Demonstrate a VQE workflow for hydrogen.

11. ستخزن النتائج الكمومية في SQL لتكون مرجعًا دائمًا.

11. Store quantum results in SQL as persistent experimental records.

لماذا نستخدم مثال الجدول الدوري هنا بدلًا من تحميل جينوم كامل؟ لأن تدفق بيانات جينوم واحد يمكن أن يصبح ضخمًا عند جمع القراءات والتجميعات والتعليقات والميزات ونتائج التحليل. لذلك نستخدم هذا المشروع مثالًا مضبوط الحجم يوضح كيف يمكن أن تعمل OOP وSQL والمصفوفات والذكاء الاصطناعي والحوسبة الكمومية معًا. وعندما تكون السرعة ذات أثر على قرار صحي أو بحثي، تصبح قابلية التتبع والتحقق والقيود الأخلاقية جزءًا من هندسة النظام، لا إضافات فلسفية منفصلة.

Why use the periodic table instead of loading an entire genome here? A single genomic data stream can become enormous when reads, assemblies, annotations, features, and analysis results are combined. This project therefore uses a controlled-size example to demonstrate how OOP, SQL, numerical matrices, AI, and quantum computing can work together. When computational speed can affect a health or research decision, traceability, validation, and ethical constraints are part of system engineering rather than separate philosophical additions.

الباب الأول: الجدول الدوري كبيانات

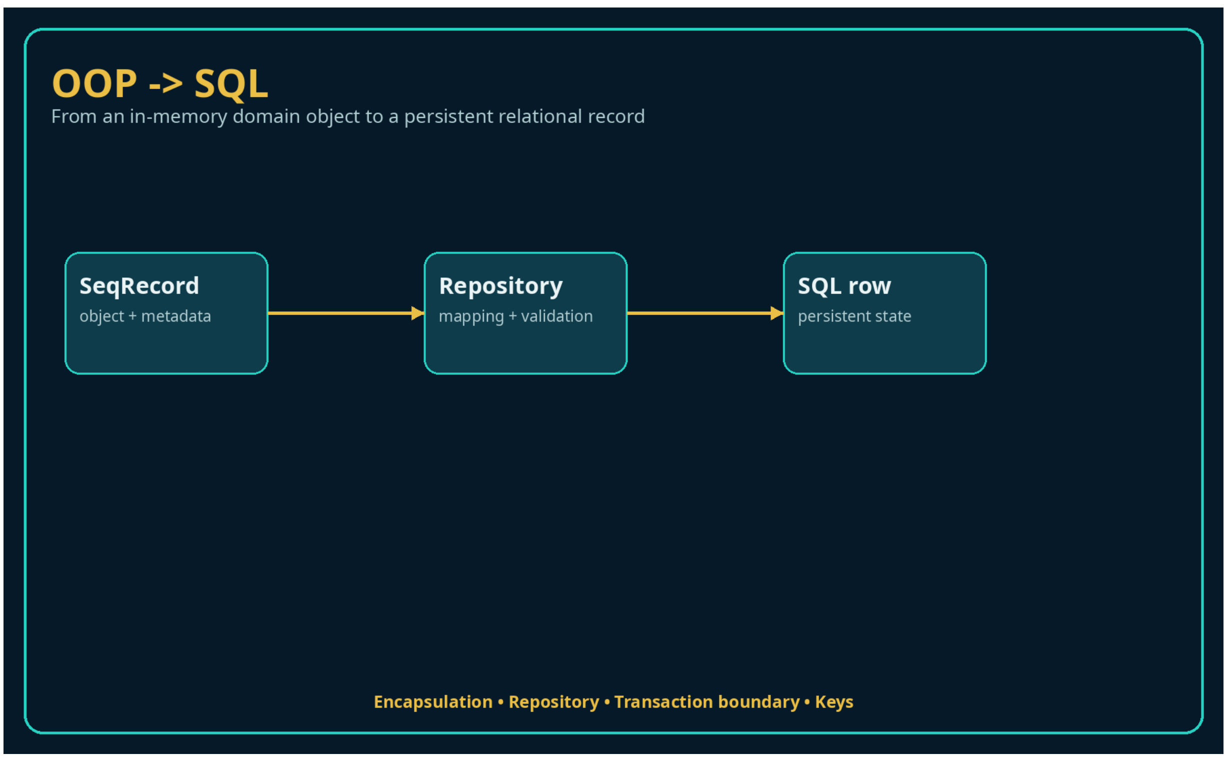
Part I: The Periodic Table as Data

الفصل الأول: بناء مخطط SQL للجدول الدوري – SQL Schema Design

Chapter 1: SQL Schema Design

من OOP إلى SQL

From OOP to SQL



توضح اللوحة كيف ينتقل الكائن الممثل بخصائصه وسلوكه إلى صف علائقي مع Repository وحدود معاملة ومفاتيح قاعدة البيانات.

This plate shows how an object with state and behavior is mapped into a relational row through a repository, transaction boundary, and database keys.

1.1 مقدمة

Introduction

في هذا الفصل، سنحوّل الجدول الدوري من تمثيل جدولي إلى قاعدة بيانات علائقية يمكن الاستعلام عنها وتحديثها والتحقق من سلامتها وتحليلها. الفكرة ليست أن نستبدل الكيمياء بقاعدة بيانات، بل أن نرى كيف تساعدنا OOP وSQL على تمثيل بيانات علمية منظمة بطريقة قابلة لإعادة الاستخدام والتدقيق.

In this chapter, the periodic table is transformed from a printed chart into a relational database that can be queried, updated, validated, and analyzed. This is the first step toward making the table a living computational resource.

1.2 مخطط قاعدة البيانات

Database Schema

سنقوم بتصميم قاعدة بيانات تحتوي على الجداول التالية:

We will design a relational database containing the following tables:

1. elements – الجدول الرئيسي للعناصر.

1. elements — the main element table.

2. groups – جدول المجموعات (18 مجموعة).

2. groups — the 18 periodic-table groups.

3. periods – جدول الدورات (7 دورات).

3. periods — the 7 periods.

4. categories – جدول الفئات (فلزات، لافلزات، إلخ).

4. categories — broad element categories such as metals and nonmetals.

5. properties – جدول الخواص الإضافية (اختياري للتوسع).

5. properties — an optional extension table for additional properties.

العلاقات بين الجداول:

The relationships between these tables are:

· elements → groups (علاقة N:1)

• elements → groups (N:1 relationship)

· elements → periods (علاقة N:1)

• elements → periods (N:1 relationship)

· elements → categories (علاقة N:1)

• elements → categories (N:1 relationship)

· elements → properties (علاقة N:1) – للتوسع المستقبلي

• properties → elements (N:1 relationship) — reserved for future extension.

1.3 كود إنشاء الجداول

Table Creation Code

SQL

```
PRAGMA foreign_keys = ON;

CREATE TABLE groups (
    group_number INTEGER PRIMARY KEY,
    group_name TEXT NOT NULL,
    group_english TEXT NOT NULL
);

CREATE TABLE periods (
    period_number INTEGER PRIMARY KEY,
    period_name TEXT NOT NULL
);

CREATE TABLE categories (
    category_id INTEGER PRIMARY KEY AUTOINCREMENT,
    category_name TEXT NOT NULL,
    category_english TEXT UNIQUE NOT NULL
);

CREATE TABLE elements (
    atomic_number INTEGER PRIMARY KEY,
    symbol TEXT UNIQUE NOT NULL,
    name_arabic TEXT NOT NULL,
```

```
    name_english TEXT NOT NULL,
    atomic_weight REAL NOT NULL,
    group_number INTEGER NOT NULL,
    period INTEGER NOT NULL,
    category_id INTEGER NOT NULL,
    state TEXT,
    density REAL,
    melting_point REAL,
    boiling_point REAL,
    electron_configuration TEXT,
    oxidation_states TEXT,
    electronegativity REAL,
    atomic_radius REAL,
    ionization_energy REAL,
    discovery_year INTEGER,
    is_radioactive INTEGER NOT NULL DEFAULT 0,
    is_synthetic INTEGER NOT NULL DEFAULT 0,
    FOREIGN KEY (group_number) REFERENCES groups(group_number),
    FOREIGN KEY (period) REFERENCES periods(period_number),
    FOREIGN KEY (category_id) REFERENCES categories(category_id)
);

CREATE INDEX idx_elements_symbol ON elements(symbol);
CREATE INDEX idx_elements_group ON elements(group_number);
CREATE INDEX idx_elements_period ON elements(period);
CREATE INDEX idx_elements_category ON elements(category_id);
CREATE INDEX idx_elements_atomic_weight ON elements(atomic_weight);

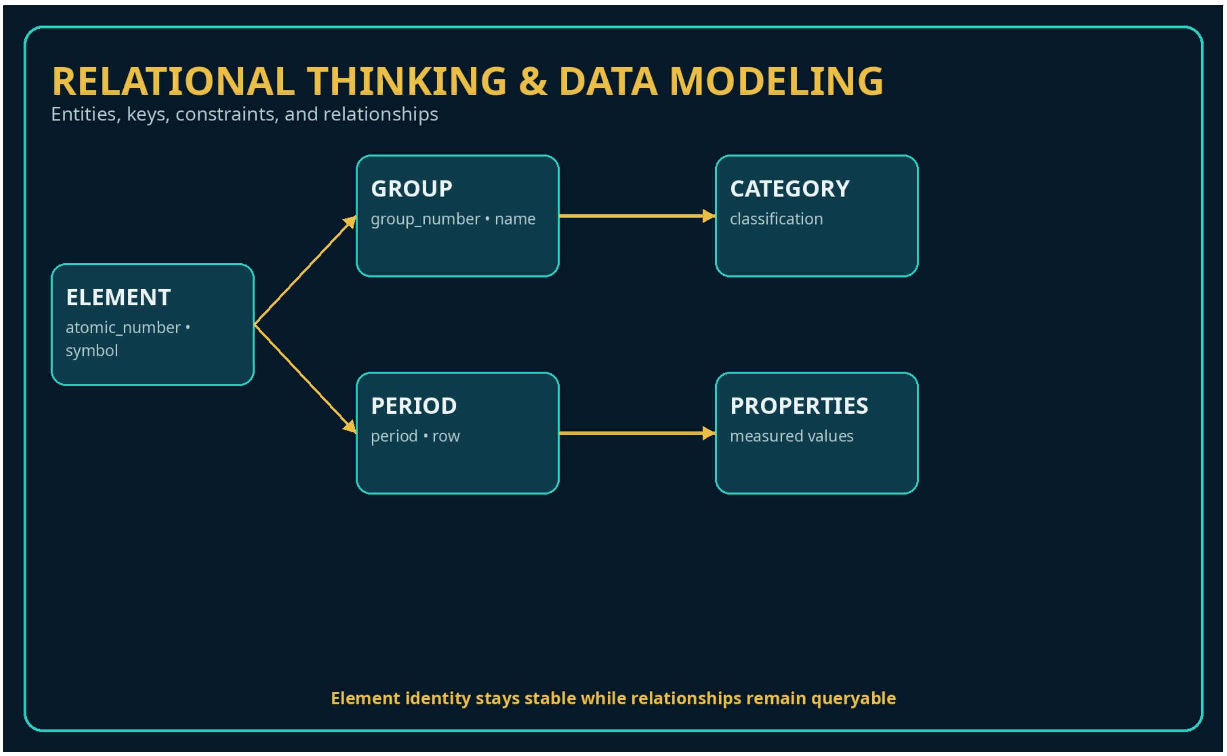
INSERT INTO groups(group_number, group_name, group_english) VALUES
(1, 'الفلزات القلوية', 'Alkali metals'),
(2, 'الفلزات القلوية الترابية', 'Alkaline earth metals'),
(3, 'مجموعة السكانديوم', 'Scandium group'),
(4, 'مجموعة التيتانيوم', 'Titanium group'),
(5, 'مجموعة الفاناديوم', 'Vanadium group'),
(6, 'مجموعة الكروم', 'Chromium group'),
(7, 'مجموعة المنغنيز', 'Manganese group'),
(8, 'مجموعة الحديد', 'Iron group'),
(9, 'مجموعة الكوبالت', 'Cobalt group'),
(10, 'مجموعة النيكل', 'Nickel group'),
(11, 'مجموعة النقود', 'Coinage metals'),
(12, 'مجموعة الزنك', 'Zinc group'),
(13, 'مجموعة البورون', 'Boron group'),
(14, 'مجموعة الكربون', 'Carbon group'),
(15, 'مجموعة النيتروجين', 'Nitrogen group'),
(16, 'مجموعة الكالكوجينات', 'Chalcogens'),
(17, 'الهالوجينات', 'Halogens'),
(18, 'الغازات النبيلة', 'Noble gases');
INSERT INTO periods(period_number, period_name) VALUES
(1, 'الدورة 1'),
(2, 'الدورة 2'),
(3, 'الدورة 3'),
(4, 'الدورة 4'),
(5, 'الدورة 5'),
(6, 'الدورة 6'),
(7, 'الدورة 7');
INSERT INTO categories(category_name, category_english) VALUES
('فلزات قلوية', 'Alkali'),
('فلزات قلوية ترابية', 'Alkaline'),
('فلزات انتقالية', 'Transition'),
('عناصر الفئة p', 'p-block'),
('عناصر الفئة f', 'f-block'),
('هالوجينات', 'Halogen'),
('غازات نبيلة', 'Noble gas');
```

1.4 شرح المخطط

Schema Explanation

التفكير العلائقي ونموذج البيانات

Relational Thinking and Data Modeling



توضح اللوحة العلاقات بين الكيانات والعلاقات التي تجعل مخطط الجدول الدوري قابلًا للاستعلام والتوسع بدل تخزين البيانات في جدول مسطح واحد.

This plate shows how entities and relationships turn the periodic-table data into an extensible relational model rather than a single flat table.

يمكنك أن تنظر إلى هذا المخطط باعتباره مثالًا مباشرًا على ترجمة نموذج كائني إلى تصميم SQL: للكائن هوية وخصائص، وللمستودع مسؤولية الوصول إلى البيانات، ولقيود SQL دور في حماية الاتساق.

Dear reader, this schema is not merely a collection of SQL statements. It is a data model that expresses the logical structure of the periodic table.

دعنا نفهم كل جدول:

Let us understand the role of each table:

· جدول groups: يمثل الأعمدة الـ 18 في الجدول الدوري. كل مجموعة تحتوي على عناصر متشابهة في الخواص الكيميائية (مثل الفلزات القلوية، الغازات النبيلة). ربطنا كل عنصر بمجموعته لتسهيل الاستعلامات حسب التشابه الكيميائي.

The groups table represents the 18 columns of the periodic table. Elements in a group share important chemical patterns, and the foreign-key relationship makes group-based queries explicit.

· جدول periods: يمثل الصفوف الـ 7 (الدورات). كل دورة تمثل مستوى طاقة رئيسيًا. هذا الجدول يساعد في فهم توزيع الإلكترونات واستقرار الذرات.

The periods table represents the 7 rows, with each period corresponding to a principal electron-energy level.

· جدول categories: يمثل التصنيفات العامة مثل "فلزات"، "لافلزات"، "أشباه فلزات". هذا التصنيف أساسي لفهم سلوك العناصر.

The categories table stores broad classifications such as metals, nonmetals, and metalloids; these labels are useful for exploratory analysis but do not replace a detailed chemical taxonomy.

· جدول elements: هو قلب قاعدة البيانات. يحتوي على جميع بيانات العناصر الـ 118. لاحظ أننا وضعنا أنواع بيانات دقيقة لكل خاصية: REAL للأرقام العشرية، INTEGER للأعداد الصحيحة، TEXT للنصوص، وBOOLEAN للقيم المنطقية. هذا يضمن دقة البيانات وسلامتها.

The elements table is the core dataset. It stores the 118 element identities and their numerical properties using explicit data types.

· جدول properties: هذا جدول للتوسع المستقبلي. يمكنك إضافة أي خاصية جديدة لأي عنصر دون تغيير هيكل الجدول الأساسي، مما يجعل قاعدة البيانات مرنة وقابلة للتطوير.

The properties table is reserved for future extensions; new element properties can be added without changing the core identity table.

1.5 القيود (Constraints)

Constraints

ضمان سلامة البيانات

Data Integrity Assurance

قاعدة البيانات ليست وعاءً محايدًا فقط؛ تصميم القيود فيها يحدد ما الذي يمكن قبوله وما الذي يجب رفضه. لذلك نستخدم القيود هنا لحماية سلامة بيانات المثال، لا كزينة في مخطط SQL.

A database is not only a container for data; its constraints can act as guardians of integrity. The following constraints help keep inserted values valid and coherent:.

القيد الغرض

Constraint | Purpose

PRIMARY KEY يمنع تكرار الأعداد الذرية.

PRIMARY KEY | Prevents duplicate atomic numbers.

UNIQUE يمنع تكرار الرموز الكيميائية

UNIQUE | Prevents duplicate chemical symbols.

NOT NULL يمنع إدخال بيانات فارغة للحقول الأساسية.

NOT NULL | Prevents missing values in required fields.

CHECK يتحقق من صحة الحالة (صلب/سائل/غاز).

CHECK | Restricts a field to an allowed state.

FOREIGN KEY يضمن أن كل عنصر يشير إلى مجموعة ودورة وفئة موجودة.

FOREIGN KEY | Ensures that referenced groups, periods, and categories exist.

ON DELETE CASCADE عند حذف مجموعة، تُحذف عناصرها تلقائيًا.

ON DELETE CASCADE automatically removes dependent rows when a referenced group is deleted; use this rule only when the deletion semantics are scientifically and operationally appropriate.

1.6 لماذا هذه التصميمات؟

Why This Design?

هذا التصميم يحقق عدة أهداف تعليمية وهندسية:

This design serves several educational and engineering goals:

1. التطبيع (Normalization): فصل البيانات في جداول منفصلة يقلل التكرار ويحسن الأداء.

1. Normalization: separating responsibilities reduces duplication and improves maintainability.

2. المرونة: جدول properties يسمح بإضافة خواص جديدة دون تعديل الهيكل الأساسي.

2. Flexibility: the extension table can hold additional properties without redesigning the core identity table.

3. قابلية التوسع: يمكن ربط هذا المخطط بجداول أخرى (مثل الجزيئات، التفاعلات الكيميائية، البيانات الطيفية).

3. Scalability: the schema can be linked to molecules, reactions, spectra, or other scientific datasets.

4. التوافق مع الأجزاء القادمة: هذا المخطط مهيأ لاستخراج المصفوفات، وتدريب نماذج الذكاء الاصطناعي، ومحاكاة الخواص الكمومية.

4. Forward compatibility: the structure is prepared for matrix extraction, machine learning, and quantum representations.

1.7 تمارين وتطبيقات

Exercises

1. أضف عمودًا جديدًا إلى جدول elements يسمى crystal_structure من نوع TEXT.

1. Add a crystal_structure TEXT column to the elements table.

2. أنشئ جدولًا لتخزين نظائر العناصر (isotopes) مرتبطًا بـ elements عبر atomic_number.

2. Create an isotope table linked to elements through atomic_number.

3. اكتب استعلامًا يعرض جميع العناصر التي اكتشفت بعد عام 2000.

3. Write a query that lists elements discovered after the year 2000.

4. أنشئ فهرسًا على عمود atomic_weight لتسريع استعلامات الترتيب حسب الوزن.

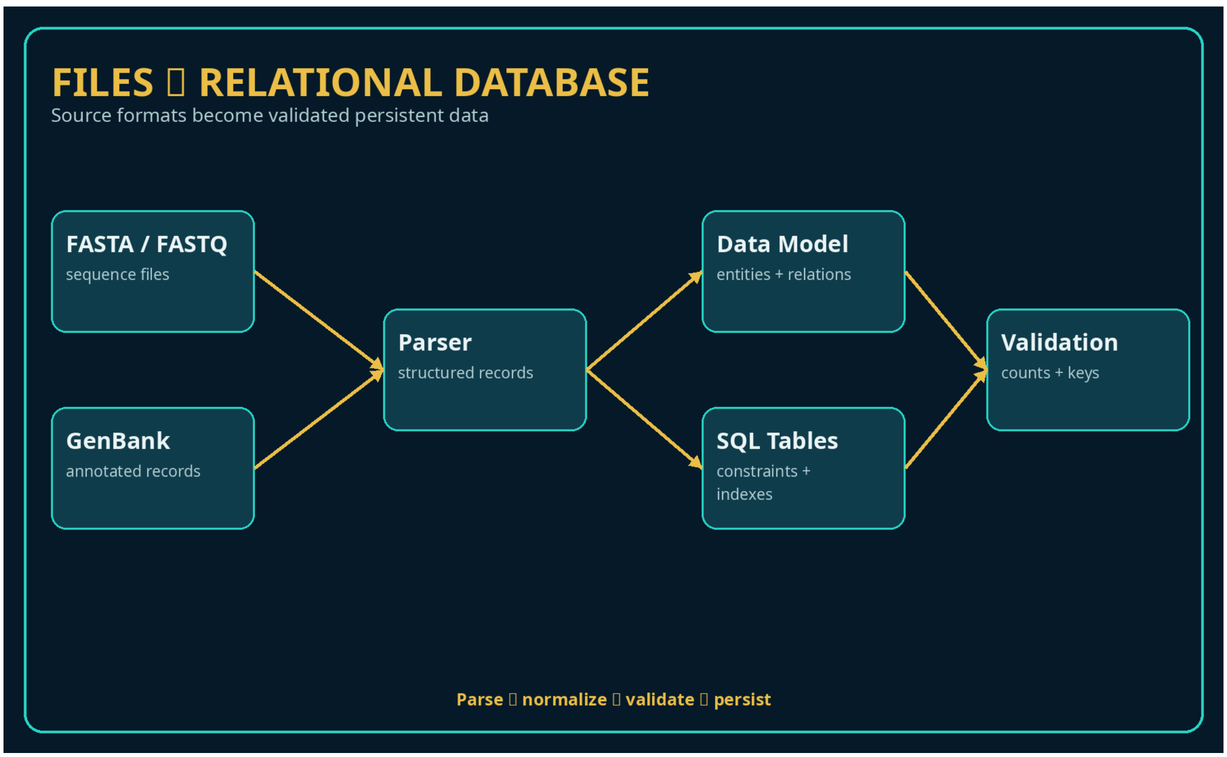
4. Create an index on atomic_weight to accelerate ordered searches.

الفصل الثاني: إدخال بيانات العناصر الـ 118 – Inserting 118 Elements

Chapter 2: Inserting the 118 Elements

من الملفات إلى قاعدة بيانات علائقية.

From Files to a Relational Database.



توضح اللوحة انتقال البيانات من ملفات المصدر مثل FASTA وGenBank إلى نموذج منظم يحافظ على الهوية والعلاقات والمفاتيح الخارجية.

This plate shows the transformation from source files such as FASTA and GenBank into a structured relational database that preserves identity, relationships, and foreign keys.

2.1 مقدمة

Introduction

بعد بناء المخطط، تأتي المرحلة التي تجعل القاعدة قابلة للاستخدام فعليًا: إدخال العناصر الـ118 مع الحفاظ على الهوية والمصدر والوحدة ونوع البيانات. لا يكفي أن نملأ الصفوف؛ يجب أن نعرف من أين جاءت القيمة وكيف نتحقق منها قبل أن تصبح جزءًا من النظام.

After the schema is defined, the database becomes useful only when the 118 elements are loaded with stable identities, provenance, units, and appropriate data types. Loading rows is not enough; every value should have a source and a validation rule before it becomes part of the system.

2.2 مبدأ الإدخال القابل لإعادة الإنتاج

Reproducible Loading

سنفصل بين البيانات الخام، وعملية التحويل، ومرحلة الإدخال. هذا الفصل يمنع أن تصبح قاعدة البيانات نسخة غامضة من ملف لا نعرف متى أو كيف أُنشئ.

Separate raw data, transformation, and database loading. This prevents the database from becoming an opaque copy of an unknown file and makes the pipeline repeatable.

2.3 إدخال جماعي باستخدام SQL

Bulk SQL Loading

يمكن توليد أوامر الإدخال من مصدر منظم، ثم تنفيذها داخل معاملة واحدة. إذا فشل صف واحد، يمكن التراجع عن العملية بدل ترك قاعدة البيانات في حالة نصف مكتملة.

Generate INSERT statements from a structured source and execute them inside a transaction. If validation fails, the transaction can be rolled back rather than leaving a partially populated database.

SQL

```
BEGIN TRANSACTION;
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (1, 'H', 'هيدروجين', 'Hydrogen', 1.008, 1, 1, (SELECT category_id FROM
categories WHERE category_english = 'Alkali'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (2, 'He', 'هيليوم', 'Helium', 4.003, 1, 18, (SELECT category_id FROM
categories WHERE category_english = 'Noble gas'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (3, 'Li', 'ليثيوم', 'Lithium', 6.941, 2, 1, (SELECT category_id FROM
categories WHERE category_english = 'Alkali'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (4, 'Be', 'بيريليوم', 'Beryllium', 9.012, 2, 2, (SELECT category_id FROM
categories WHERE category_english = 'Alkaline'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (5, 'B', 'بورون', 'Boron', 10.812, 2, 13, (SELECT category_id FROM
categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (6, 'C', 'كربون', 'Carbon', 12.011, 2, 14, (SELECT category_id FROM
categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (7, 'N', 'نيتروجين', 'Nitrogen', 14.007, 2, 15, (SELECT category_id FROM
categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (8, 'O', 'أكسجين', 'Oxygen', 15.999, 2, 16, (SELECT category_id FROM
categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (9, 'F', 'فلور', 'Fluorine', 18.998, 2, 17, (SELECT category_id FROM
categories WHERE category_english = 'Halogen'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (10, 'Ne', 'نيون', 'Neon', 20.18, 2, 18, (SELECT category_id FROM
categories WHERE category_english = 'Noble gas'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (11, 'Na', 'صوديوم', 'Sodium', 22.99, 3, 1, (SELECT category_id FROM
categories WHERE category_english = 'Alkali'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (12, 'Mg', 'مغنيسيوم', 'Magnesium', 24.305, 3, 2, (SELECT category_id FROM
categories WHERE category_english = 'Alkaline'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (13, 'Al', 'ألومنيوم', 'Aluminium', 26.982, 3, 13, (SELECT category_id FROM
categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (14, 'Si', 'سيليكون', 'Silicon', 28.086, 3, 14, (SELECT category_id FROM
categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (15, 'P', 'فوسفور', 'Phosphorus', 30.974, 3, 15, (SELECT category_id FROM
categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (16, 'S', 'كبريت', 'Sulfur', 32.067, 3, 16, (SELECT category_id FROM
categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (17, 'Cl', 'كلور', 'Chlorine', 35.453, 3, 17, (SELECT category_id FROM
categories WHERE category_english = 'Halogen'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (18, 'Ar', 'أرجون', 'Argon', 39.948, 3, 18, (SELECT category_id FROM
categories WHERE category_english = 'Noble gas'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (19, 'K', 'بوتاسيوم', 'Potassium', 39.098, 4, 1, (SELECT category_id FROM
categories WHERE category_english = 'Alkali'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (20, 'Ca', 'كالسيوم', 'Calcium', 40.078, 4, 2, (SELECT category_id FROM
categories WHERE category_english = 'Alkaline'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (21, 'Sc', 'سكانديوم', 'Scandium', 44.956, 4, 3, (SELECT category_id FROM
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (22, 'Ti', 'تيتانيوم', 'Titanium', 47.867, 4, 4, (SELECT category_id FROM
```

```
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (23, 'V', 'فاناديوم', 'Vanadium', 50.944, 4, 5, (SELECT category_id FROM categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (24, 'Cr', 'كروم', 'Chromium', 51.996, 4, 6, (SELECT category_id FROM categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (25, 'Mn', 'منغنيز', 'Manganese', 54.938, 4, 7, (SELECT category_id FROM categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (26, 'Fe', 'حديد', 'Iron', 55.845, 4, 8, (SELECT category_id FROM categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (27, 'Co', 'كوبالت', 'Cobalt', 58.933, 4, 9, (SELECT category_id FROM categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (28, 'Ni', 'نيكل', 'Nickel', 58.693, 4, 10, (SELECT category_id FROM categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (29, 'Cu', 'نحاس', 'Copper', 63.546, 4, 11, (SELECT category_id FROM categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (30, 'Zn', 'زنك', 'Zinc', 65.39, 4, 12, (SELECT category_id FROM categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (31, 'Ga', 'غاليوم', 'Gallium', 69.723, 4, 13, (SELECT category_id FROM categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (32, 'Ge', 'جرمانيوم', 'Germanium', 72.61, 4, 14, (SELECT category_id FROM categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (33, 'As', 'زرنيخ', 'Arsenic', 74.922, 4, 15, (SELECT category_id FROM categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (34, 'Se', 'سيلينيوم', 'Selenium', 78.96, 4, 16, (SELECT category_id FROM categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (35, 'Br', 'بروم', 'Bromine', 79.904, 4, 17, (SELECT category_id FROM categories WHERE category_english = 'Halogen'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (36, 'Kr', 'كريبتون', 'Krypton', 83.8, 4, 18, (SELECT category_id FROM categories WHERE category_english = 'Noble gas'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (37, 'Rb', 'روبيديوم', 'Rubidium', 85.468, 5, 1, (SELECT category_id FROM categories WHERE category_english = 'Alkali'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (38, 'Sr', 'سترونشيوم', 'Strontium', 87.62, 5, 2, (SELECT category_id FROM categories WHERE category_english = 'Alkaline'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (39, 'Y', 'إيتريوم', 'Yttrium', 88.906, 5, 3, (SELECT category_id FROM categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (40, 'Zr', 'زركونيوم', 'Zirconium', 91.224, 5, 4, (SELECT category_id FROM categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (41, 'Nb', 'نيوبيوم', 'Niobium', 92.906, 5, 5, (SELECT category_id FROM categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (42, 'Mo', 'موليبدينوم', 'Molybdenum', 95.94, 5, 6, (SELECT category_id FROM categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (43, 'Tc', 'تكنيتيوم', 'Technetium', 98, 5, 7, (SELECT category_id FROM categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (44, 'Ru', 'روثينيوم', 'Ruthenium', 101.07, 5, 8, (SELECT category_id FROM
```

```
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (45, 'Rh', 'روديوم', 'Rhodium', 102.906, 5, 9, (SELECT category_id FROM
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (46, 'Pd', 'بلاديوم', 'Palladium', 106.42, 5, 10, (SELECT category_id FROM
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (47, 'Ag', 'فضة', 'Silver', 107.868, 5, 11, (SELECT category_id FROM
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (48, 'Cd', 'كادميوم', 'Cadmium', 112.412, 5, 12, (SELECT category_id FROM
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (49, 'In', 'إنديوم', 'Indium', 114.818, 5, 13, (SELECT category_id FROM
categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (50, 'Sn', 'قصدير', 'Tin', 118.711, 5, 14, (SELECT category_id FROM
categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (51, 'Sb', 'أنتيمون', 'Antimony', 121.76, 5, 15, (SELECT category_id FROM
categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (52, 'Te', 'تيلوريوم', 'Tellurium', 127.6, 5, 16, (SELECT category_id FROM
categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (53, 'I', 'يود', 'Iodine', 126.904, 5, 17, (SELECT category_id FROM
categories WHERE category_english = 'Halogen'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (54, 'Xe', 'زينون', 'Xenon', 131.29, 5, 18, (SELECT category_id FROM
categories WHERE category_english = 'Noble gas'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (55, 'Cs', 'سيزيوم', 'Caesium', 132.905, 6, 1, (SELECT category_id FROM
categories WHERE category_english = 'Alkali'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (56, 'Ba', 'باريوم', 'Barium', 137.328, 6, 2, (SELECT category_id FROM
categories WHERE category_english = 'Alkaline'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (57, 'La', 'لانثانوم', 'Lanthanum', 138.906, 6, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (58, 'Ce', 'سيريوم', 'Cerium', 140.116, 6, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (59, 'Pr', 'براسيوديميوم', 'Praseodymium', 140.908, 6, 3, (SELECT category_id
FROM categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (60, 'Nd', 'نيوديميوم', 'Neodymium', 144.24, 6, 3, (SELECT category_id
FROM categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (61, 'Pm', 'بروميثيوم', 'Promethium', 145, 6, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (62, 'Sm', 'ساماريوم', 'Samarium', 150.36, 6, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (63, 'Eu', 'يوروبيوم', 'Europium', 151.964, 6, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (64, 'Gd', 'جادولينيوم', 'Gadolinium', 157.25, 6, 3, (SELECT category_id
FROM categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (65, 'Tb', 'تيربيوم', 'Terbium', 158.925, 6, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (66, 'Dy', 'ديسبروسيوم', 'Dysprosium', 162.5, 6, 3, (SELECT category_id
```

```
FROM categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (67, 'Ho', 'هولميوم', 'Holmium', 164.93, 6, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (68, 'Er', 'إربيوم', 'Erbium', 167.26, 6, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (69, 'Tm', 'ثوليوم', 'Thulium', 168.934, 6, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (70, 'Yb', 'إيتربيوم', 'Ytterbium', 173.04, 6, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (71, 'Lu', 'لوتيتيوم', 'Lutetium', 174.967, 6, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (72, 'Hf', 'هافنيوم', 'Hafnium', 178.49, 6, 4, (SELECT category_id FROM
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (73, 'Ta', 'تانتالوم', 'Tantalum', 180.948, 6, 5, (SELECT category_id FROM
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (74, 'W', 'تنغستن', 'Tungsten', 183.84, 6, 6, (SELECT category_id FROM
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (75, 'Re', 'رينيوم', 'Rhenium', 186.207, 6, 7, (SELECT category_id FROM
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (76, 'Os', 'أوزميوم', 'Osmium', 190.23, 6, 8, (SELECT category_id FROM
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (77, 'Ir', 'إيريديوم', 'Iridium', 192.217, 6, 9, (SELECT category_id FROM
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (78, 'Pt', 'بلاتين', 'Platinum', 195.078, 6, 10, (SELECT category_id FROM
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (79, 'Au', 'ذهب', 'Gold', 196.967, 6, 11, (SELECT category_id FROM
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (80, 'Hg', 'زئبق', 'Mercury', 200.59, 6, 12, (SELECT category_id FROM
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (81, 'Tl', 'ثاليوم', 'Thallium', 204.383, 6, 13, (SELECT category_id FROM
categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (82, 'Pb', 'رصاص', 'Lead', 207.2, 6, 14, (SELECT category_id FROM
categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (83, 'Bi', 'بزموت', 'Bismuth', 208.98, 6, 15, (SELECT category_id FROM
categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (84, 'Po', 'بولونيوم', 'Polonium', 209, 6, 16, (SELECT category_id FROM
categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (85, 'At', 'أستاتين', 'Astatine', 210, 6, 17, (SELECT category_id FROM
categories WHERE category_english = 'Halogen'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (86, 'Rn', 'رادون', 'Radon', 222, 6, 18, (SELECT category_id FROM
categories WHERE category_english = 'Noble gas'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (87, 'Fr', 'فرانسيوم', 'Francium', 223, 7, 1, (SELECT category_id FROM
categories WHERE category_english = 'Alkali'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (88, 'Ra', 'راديوم', 'Radium', 226, 7, 2, (SELECT category_id FROM
```

```
categories WHERE category_english = 'Alkaline'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (89, 'Ac', 'أكتينيوم', 'Actinium', 227, 7, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (90, 'Th', 'ثوريوم', 'Thorium', 232.038, 7, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (91, 'Pa', 'بروتكتينيوم', 'Protactinium', 231.036, 7, 3, (SELECT category_id
FROM categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (92, 'U', 'يورانيوم', 'Uranium', 238.029, 7, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (93, 'Np', 'نبتونيوم', 'Neptunium', 237, 7, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (94, 'Pu', 'بلوتونيوم', 'Plutonium', 244, 7, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (95, 'Am', 'أمريسيوم', 'Americium', 243, 7, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (96, 'Cm', 'كوريوم', 'Curium', 247, 7, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (97, 'Bk', 'بركيليوم', 'Berkelium', 247, 7, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (98, 'Cf', 'كاليفورنيوم', 'Californium', 251, 7, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (99, 'Es', 'أينشتاينيوم', 'Einsteinium', 252, 7, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (100, 'Fm', 'فرميوم', 'Fermium', 257, 7, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (101, 'Md', 'مندليفيوم', 'Mendelevium', 258, 7, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (102, 'No', 'نوبليوم', 'Nobelium', 259, 7, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (103, 'Lr', 'لورنسيوم', 'Lawrencium', 262, 7, 3, (SELECT category_id FROM
categories WHERE category_english = 'f-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (104, 'Rf', 'رذرفورديوم', 'Rutherfordium', 267, 7, 4, (SELECT category_id
FROM categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (105, 'Db', 'دوبنيوم', 'Dubnium', 268, 7, 5, (SELECT category_id FROM
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (106, 'Sg', 'سيبورغيوم', 'Seaborgium', 269, 7, 6, (SELECT category_id FROM
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (107, 'Bh', 'بوهريوم', 'Bohrium', 270, 7, 7, (SELECT category_id FROM
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (108, 'Hs', 'هاسيوم', 'Hassium', 269, 7, 8, (SELECT category_id FROM
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (109, 'Mt', 'مايتنريوم', 'Meitnerium', 278, 7, 9, (SELECT category_id FROM
categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period,
group_number, category_id) VALUES (110, 'Ds', 'دارمشتاتيوم', 'Darmstadtium', 281, 7, 10, (SELECT category_id
```

```
FROM categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (111, 'Rg', 'رونتجينيوم', 'Roentgenium', 281, 7, 11, (SELECT category_id FROM categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (112, 'Cn', 'كوبرنيسيوم', 'Copernicium', 285, 7, 12, (SELECT category_id FROM categories WHERE category_english = 'Transition'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (113, 'Nh', 'نيهونيوم', 'Nihonium', 284, 7, 13, (SELECT category_id FROM categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (114, 'Fl', 'فليروفيوم', 'Flerovium', 289, 7, 14, (SELECT category_id FROM categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (115, 'Mc', 'موسكوفيوم', 'Moscovium', 288, 7, 15, (SELECT category_id FROM categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (116, 'Lv', 'ليفرموريوم', 'Livermorium', 293, 7, 16, (SELECT category_id FROM categories WHERE category_english = 'p-block'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (117, 'Ts', 'تنسين', 'Tennessine', 292, 7, 17, (SELECT category_id FROM categories WHERE category_english = 'Halogen'));
INSERT INTO elements (atomic_number, symbol, name_arabic, name_english, atomic_weight, period, group_number, category_id) VALUES (118, 'Og', 'أوغانيسون', 'Oganesson', 294, 7, 18, (SELECT category_id FROM categories WHERE category_english = 'Noble gas'));
COMMIT;
```

2.4 التحقق بعد الإدخال

Post-load Validation

يعرض هذا الاستعلام الأعمدة المطلوبة، ويرشح الصفوف بحسب الوزن الذري، ثم يرتب النتيجة من القيمة الأكبر إلى الأصغر.

After loading, verify the row count, uniqueness of atomic numbers and symbols, period and group ranges, and the absence of impossible negative atomic weights. Validation is part of the scientific method, not an optional cleanup step.

SQL

```sql
SELECT COUNT(*) AS element_count FROM elements;
SELECT COUNT(DISTINCT atomic_number) AS unique_atomic_numbers FROM elements;
SELECT COUNT(DISTINCT symbol) AS unique_symbols FROM elements;
SELECT MIN(atomic_number), MAX(atomic_number) FROM elements;
SELECT * FROM elements WHERE atomic_weight <= 0;
```

2.5 تمارين وتطبيقات

Exercises

1. أعد تشغيل عملية الإدخال مرتين، ثم صمّم القيد الذي يمنع التكرار.

1. Run the loading procedure twice and design the constraint that prevents duplicates.

2. أنشئ تقريرًا يطابق عدد العناصر مع العدد الذري الأعلى.

2. Create a validation report comparing the number of rows with the highest atomic number.

3. أضف حقلًا لمصدر القيمة source_reference، ثم اختبر استرجاع مصدر عنصر محدد.

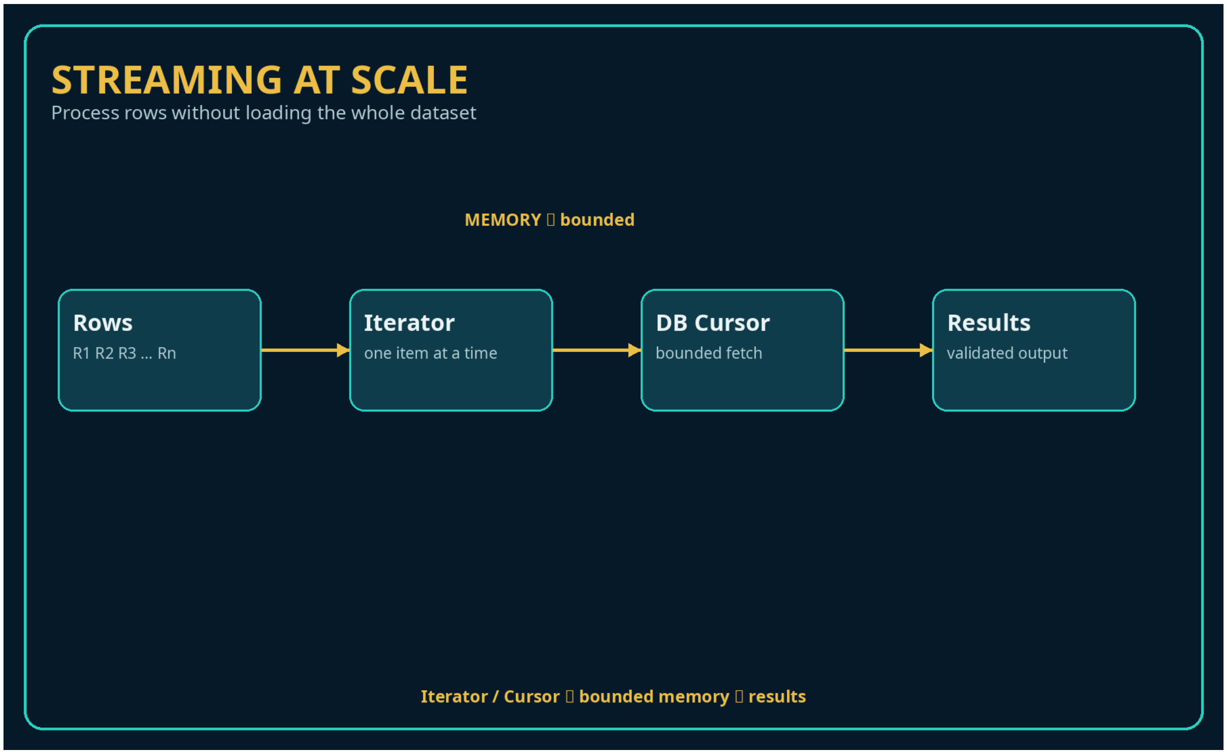
3. Add a source_reference field and test provenance retrieval for a selected element.

الفصل الثالث: الاستعلامات الأساسية – Basic Queries (SELECT, WHERE, ORDER BY)

Chapter 3: Basic Queries (SELECT, WHERE, ORDER BY)

المعالجة المتدفقة

Streaming at Scale



ونتائج الاستعلام دون تحميل كل البيانات في الذاكرة Cursor توضح اللوحة الفرق بين القراءة المتدفقة واستخدام.

This plate contrasts streaming access and cursor-based database processing so results can be handled without loading the entire dataset into memory.

3.1 القراءة تبدأ بسؤال علمي

Query Thinking

SQL ليس قائمة أوامر للحفظ؛ إنه لغة لصياغة سؤال عن البيانات. ابدأ بالسؤال: ما الكائنات التي أريد رؤيتها؟ وما الشرط الذي يحددها؟ وكيف أريد ترتيب النتيجة؟.

SQL is not a list of commands to memorize; it is a language for expressing questions about data. Start with the scientific question, then define the rows, conditions, and ordering that answer it.

3.2 SELECT وWHERE

Selecting and Filtering

SQL

```
SELECT atomic_number, symbol, name_english, atomic_weight
FROM elements
WHERE atomic_weight > 100
ORDER BY atomic_weight DESC;
```

يعرض هذا الاستعلام الأعمدة المطلوبة، ويرشح العناصر بحسب الوزن الذري، ثم يرتب النتيجة من القيمة الأعلى إلى الأدنى.

This query returns selected columns, filters rows by atomic weight, and sorts the result from the heaviest value downward.

3.3 المقارنات والنطاقات

Comparisons and Ranges

SQL

```
SELECT symbol, name_english, atomic_weight
FROM elements
WHERE atomic_weight BETWEEN 40 AND 80
ORDER BY atomic_weight;
```

يعبّر BETWEEN عن مجال شامل للحدين، وهو مفيد عند استكشاف نطاقات من الخواص المقاسة أو المجدولة.

BETWEEN expresses an inclusive range and is useful when exploring bands of measured or tabulated properties.

3.4 البحث النصي والقيم المفقودة

Text Search and NULL

SQL

```
SELECT atomic_number, symbol, name_english
FROM elements
WHERE name_english LIKE 'C%';

SELECT * FROM elements
WHERE atomic_weight IS NULL;
```

NULL للتعامل مع القيم المفقودة؛ ولا ينبغي مقارنة IS NULL مطابقة الأنماط النصية، بينما يلزم استخدام LIKE تدعم بعوامل المساواة العادية.

LIKE supports pattern matching, while IS NULL is required for missing values; NULL should never be compared with ordinary equality operators.

3.5 التجميع البسيط

Aggregation

SQL

```
SELECT c.category_english, COUNT(*) AS n
FROM elements AS e
JOIN categories AS c ON e.category_id = c.category_id
GROUP BY c.category_english
ORDER BY n DESC;
```

يربط هذا الاستعلام كل عنصر بالوصف الموثوق لمجموعته، ويتجنب تكرار بيانات المجموعة داخل كل صف عنصر.

Aggregation changes the question from “which rows?” to “what pattern appears across groups?” The interpretation remains scientific: a count describes the dataset, not a causal mechanism.

3.6 تمارين وتطبيقات

Exercises

1. اعرض عناصر المجموعة 18 مرتبة حسب العدد الذري.

1. List group-18 elements ordered by atomic number.

2. استخرج العناصر ذات الوزن الذري بين 20 و50.

2. Retrieve elements whose atomic weights fall between 20 and 50.

3. احسب عدد العناصر في كل دورة.

3. Count the elements in each period.

4. استخدم EXPLAIN QUERY PLAN لفهم كيف ينفذ SQLite الاستعلام.

4. Use EXPLAIN QUERY PLAN to inspect how SQLite executes a query.

تمنح CTE الاستعلام المعقد مرحلة وسيطة ذات اسم واضح، مما يسهل تدقيق التحويلات العلمية واختبارها.

A CTE gives a complex query a named intermediate stage, making scientific transformations easier to audit and test.

هذا الاستخدام لا يعطيك «حقيقة علمية» جديدة؛ بل يريك كيف ينفذ محرك قاعدة البيانات الاستعلام، وهو أمر مهم عند تحليل الأداء.

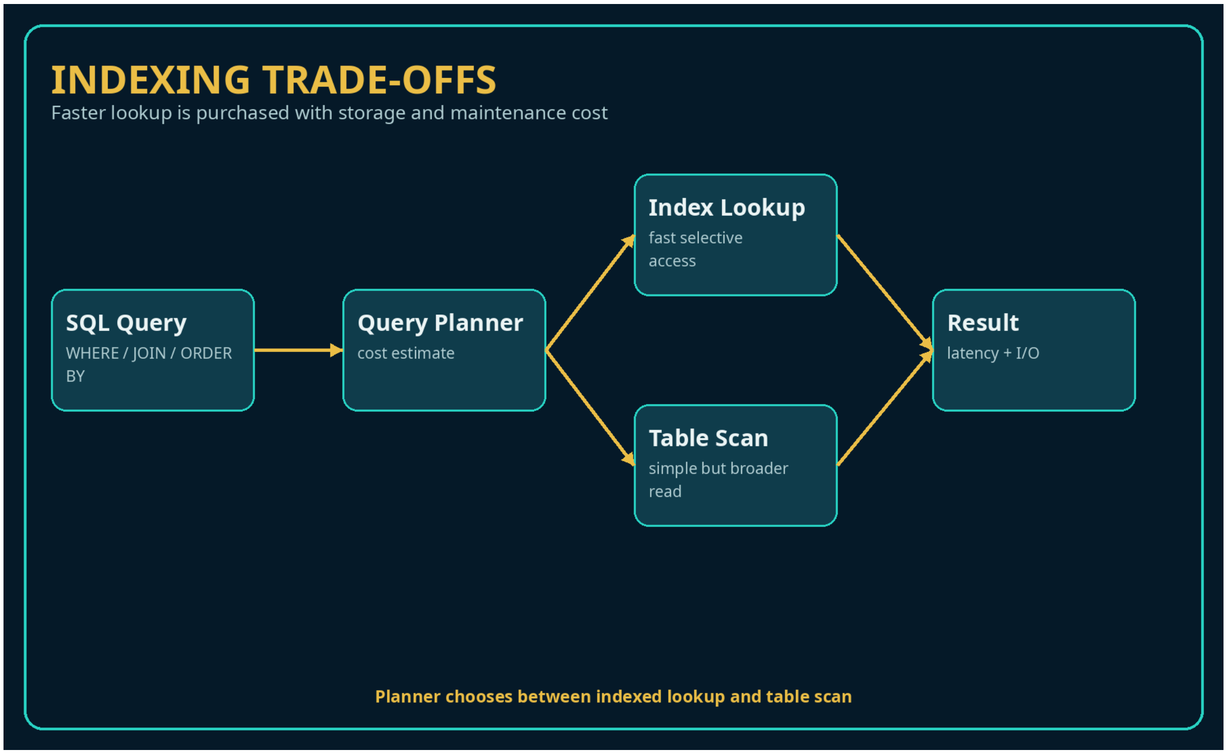
This does not create a new scientific fact; it shows how the database engine executes the query, which matters when performance must be analyzed.

الفصل الرابع: الاستعلامات المتقدمة – Advanced Queries (JOIN, GROUP BY, HAVING)

Chapter 4: Advanced Queries (JOIN, GROUP BY, HAVING)

مقايضات الفهرسة

Indexing Trade-offs



توضح اللوحة كيف يوازن تصميم الفهرس بين سرعة البحث وكلفة التخزين والتحديث، ولماذا لا يعني وجود الفهرس أن كل استعلام سيصبح أسرع.

This plate shows the trade-off between lookup speed and storage/update cost and why an index does not automatically make every query faster.

4.1 من الصف إلى العلاقة

From Rows to Relationships

حين تصبح البيانات موزعة على أكثر من جدول، تظهر قيمة العلاقات. JOIN يسمح لك بإعادة بناء الصورة العلمية من جداول صغيرة ذات مسؤوليات واضحة.

When data is distributed across normalized tables, relationships become central. JOIN reconstructs a meaningful scientific view from smaller tables with explicit responsibilities.

4.2 JOIN بين العناصر والمجموعات

Joining Elements and Groups

SQL

```
SELECT e.symbol, e.name_english, g.group_number, g.group_name
FROM elements AS e
JOIN groups AS g ON e.group_number = g.group_number
ORDER BY g.group_number, e.atomic_number;
```

يربط هذا الاستعلام كل عنصر بالوصف المعتمد لمجموعته، ويتجنب تكرار بيانات المجموعة داخل كل صف من صفوف العناصر.

The join connects each element to the authoritative description of its group and avoids duplicating group metadata in every element row.

4.3 GROUP BY وHAVING

Grouping and Filtering Groups

SQL

```
SELECT period, COUNT(*) AS n
FROM elements
GROUP BY period
HAVING COUNT(*) >= 10
ORDER BY period;
```

تعمل HAVING بعد التجميع لتصفية المجموعات، بينما تعمل WHERE قبل التجميع لتصفية الصفوف؛ وهذا الفرق أساسي عند بناء الاستعلامات العلمية.

HAVING filters groups after aggregation. This distinction matters because WHERE filters rows before grouping, whereas HAVING filters the grouped result.

4.4 CTE والاستعلامات الفرعية

CTEs and Subqueries

SQL

```
WITH heavy AS (
   SELECT * FROM elements WHERE atomic_weight > 150
)
SELECT c.category_english, COUNT(*) AS n
FROM heavy AS h
```

```
JOIN categories AS c ON h.category_id = c.category_id
GROUP BY c.category_english;
```

تمنح CTE مرحلة وسيطة مسماة داخل الاستعلام المعقد، مما يجعل التحويلات العلمية أسهل في التدقيق والاختبار.

A common table expression gives a complex query a named intermediate stage. It can make scientific transformations easier to audit and test.

4.5 تمارين وتطبيقات

Exercises

1. اربط العناصر بالدورات والمجموعات في استعلام واحد.

1. Join elements to both periods and groups in one query.

2. اعثر على الفئات التي تحتوي على أكثر من خمسة عناصر.

2. Find categories containing more than five elements.

3. اكتب CTE يحسب متوسط الوزن الذري لكل دورة.

3. Write a CTE that calculates the mean atomic weight for each period.

الباب الثاني: من SQL إلى المصفوفات

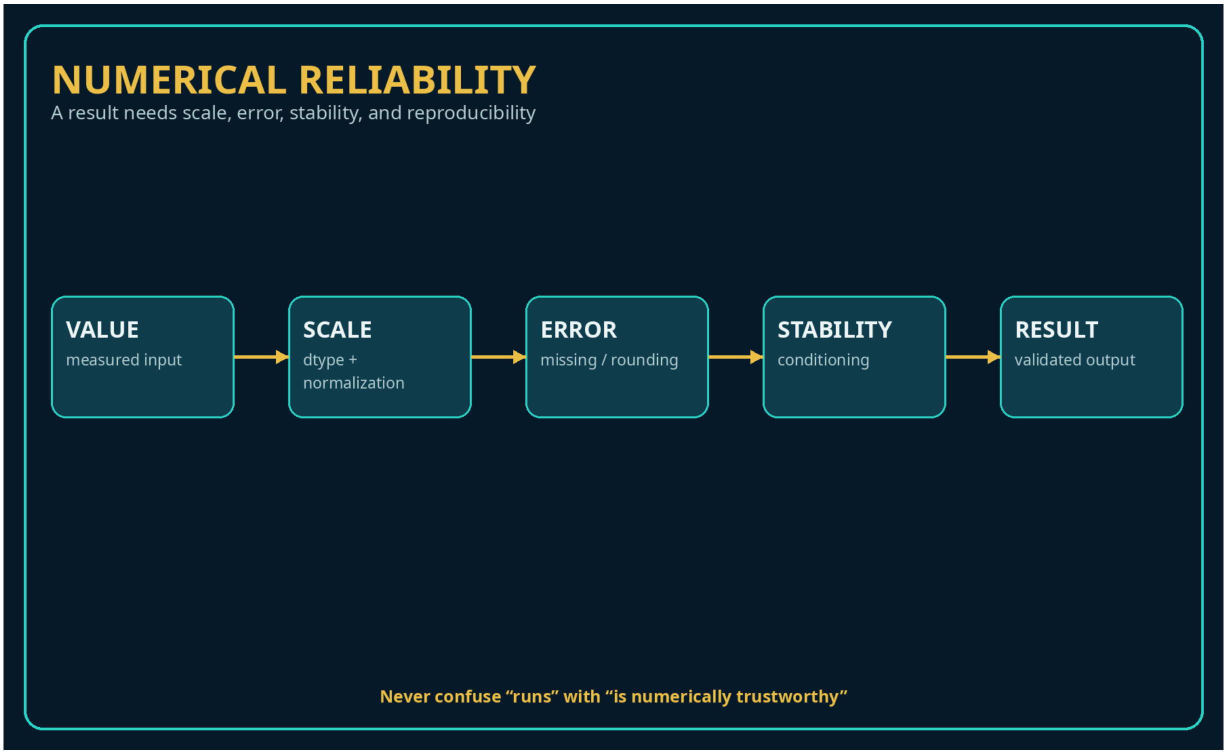
Part II: From SQL to Matrices

الفصل الخامس: استخراج المصفوفات باستخدام NumPy

Chapter 5: Matrix Extraction with NumPy

الدقة العددية

Numerical Reliability



تربط اللوحة القيمة بالمقياس والخطأ والاستقرار والنتيجة، لتذكيرك بأن المصفوفة العددية ليست موثوقة لمجرد أن الكود نفّذ دون استثناء.

This plate connects value, scale, error, stability, and result, emphasizing that numerical output is not trustworthy merely because the code ran without an exception.

المعالجة المتدفقة على نطاق واسع

Large-Scale Streaming Processing

5.1 مقدمة

Introduction

الآن بعد أن أصبح لدينا قاعدة بيانات كاملة للجدول الدوري، حان الوقت لاستخراج هذه البيانات وتحويلها إلى مصفوفات رقمية يمكن استخدامها في التحليل العددي، والتعلم الآلي، والحوسبة الكمومية.

Once the relational database exists, the next step is to extract its values into numerical matrices suitable for numerical analysis, machine learning, and later quantum representations.

في هذا الفصل، سنتعلم:

In this chapter, you will learn:

• كيف نستعلم البيانات من SQLite باستخدام Python

• Query SQLite with Python.

• كيف نحول هذه البيانات إلى مصفوفات NumPy

• Convert the rows to NumPy matrices.

• كيف نتعامل مع القيم المفقودة (NULL)

• Handle missing values (NULL/NaN).

• كيف نطبق التطبيع (Normalization) على البيانات

• Normalize numerical features.

5.2 تجهيز البيئة

Environment Setup

أولاً، تأكد من تثبيت المكتبات المطلوبة:

First, make sure the required libraries are available:

BASH

```bash
pip install numpy pandas matplotlib scikit-learn.
```

ملاحظة: وحدة sqlite3 مدمجة في Python، ولا تحتاج إلى تثبيت إضافي.

Note: Python includes sqlite3 in the standard library, so no separate installation is required.

5.3 كود استخراج المصفوفة

Matrix Extraction Code

الآن، لنكتب كودًا كاملاً لاستخراج بيانات العناصر وتحويلها إلى مصفوفة NumPy.

Now we will write a complete routine that extracts element data and converts it into a NumPy matrix.

PYTHON

```python
# ============================================================
# extract_periodic_table_matrix.py
# استخراج بيانات الجدول الدوري من SQLite وتحويلها إلى مصفوفة NumPy
# Extracting Periodic Table Data from SQLite and Converting to NumPy Matrix
# ============================================================


import sqlite3
import numpy as np
import pandas as pd
from pathlib import Path
import json


# ============================================================
# 1. الاتصال بقاعدة البيانات
# ============================================================


def get_db_connection(db_path="periodic_table.db"):
    """
    إنشاء اتصال بقاعدة البيانات مع تفعيل row_factory للوصول بالأسماء.
    Create a database connection with row_factory enabled for named access.
    """
    if not Path(db_path).exists():
        print(f"⚠️ قاعدة البيانات غير موجودة: {db_path}")
        print("⚠️ Database not found. Please create it first.")
        return None


    conn = sqlite3.connect(db_path)
    conn.row_factory = sqlite3.Row
    return conn


# ============================================================
# 2. استخراج بيانات العناصر
# ============================================================


def fetch_elements_data(conn):
    """
    استرجاع جميع بيانات العناصر من قاعدة البيانات.
    Fetch all element data from the database.
    """
    query = """
    SELECT
        e.atomic_number,
        e.symbol,
        e.name_arabic,
```

```
        e.name_english,
        e.atomic_weight,
        e.group_number,
        e.period,
        e.category_id,
        c.category_name,
        c.category_english,
        e.state,
        e.density,
        e.melting_point,
        e.boiling_point,
        e.electron_configuration,
        e.electron_shells,
        e.oxidation_states,
        e.electronegativity,
        e.atomic_radius,
        e.ionization_energy,
        e.discovery_year,
        e.is_radioactive,
        e.is_synthetic,
        e.description_arabic,
        e.description_english,
        e.color,
        e.magnetic_ordering,
        e.electrical_resistivity,
        e.thermal_conductivity,
        e.abundance_in_earth_crust,
        e.abundance_in_universe
    FROM elements e
    LEFT JOIN categories c ON e.category_id = c.category_id
    ORDER BY e.atomic_number
    """


    cursor = conn.cursor()
    cursor.execute(query)
    rows = cursor.fetchall()


    # تحويل الصفوف إلى قائمة من القواميس
    data = []
    for row in rows:
        data.append(dict(row))


    return data


# ============================================================
# 3. اختيار الخصائص العددية للمصفوفة
# ============================================================


def extract_numeric_matrix(elements_data, features=None):
    """
    استخراج مصفوفة عددية من بيانات العناصر.
    Extract a numeric matrix from element data.


    المعاملات:
        elements_data: قائمة من القواميس تحتوي على بيانات العناصر
        features: قائمة بأسماء الخصائص العددية المطلوبة (إذا كانت None، نستخدم الكل)


    Returns:
        matrix: مصفوفة NumPy (n_elements × n_features)
```

```
        feature_names: قائمة بأسماء الخصائص
        element_identifiers: قائمة بمعرفات العناصر (العدد الذري، الرمز)
"""


# الخصائص العددية الافتراضية
    default_features = [
        'atomic_weight',
        'density',
        'melting_point',
        'boiling_point',
        'electronegativity',
        'atomic_radius',
        'ionization_energy',
        'electrical_resistivity',
        'thermal_conductivity',
        'abundance_in_earth_crust',
        'abundance_in_universe'
    ]


    if features is None:
        features = default_features


    # التحقق من وجود الخصائص في البيانات
    first_row = elements_data[0]
    available_features = [f for f in features if f in first_row]


    # استخراج البيانات العددية
    matrix = []
    element_identifiers = []
    feature_names = []


    for idx, element in enumerate(elements_data):
        row = []
        for feature in available_features:
            value = element.get(feature)
            # تحويل القيم إلى float، مع معالجة القيم المفقودة
            if value is None or value == '':
                row.append(np.nan)  # استخدم NaN للقيم المفقودة
            else:
                try:
                    row.append(float(value))
                except (ValueError, TypeError):
                    row.append(np.nan)
        matrix.append(row)
        element_identifiers.append({
            'atomic_number': element['atomic_number'],
            'symbol': element['symbol'],
            'name': element['name_english']
        })


    matrix = np.array(matrix, dtype=np.float64)
    feature_names = available_features


    return matrix, feature_names, element_identifiers


# ============================================================
# 4. معالجة القيم المفقودة
# ============================================================
```

```
def handle_missing_values(matrix, strategy='mean'):
    """
    معالجة القيم المفقودة في المصفوفة.
    Handle missing values in the matrix.


    الاستراتيجيات:
        - 'mean': استبدال بالقيمة المتوسطة للعمود
        - 'median': استبدال بالقيمة الوسيطة للعمود
        - 'zero': استبدال بالصفر
        - 'drop': حذف الصفوف التي تحتوي على قيم مفقودة
        - 'interpolate': استيفاء القيم (خطي)
    """
    matrix_copy = matrix.copy()
    n_rows, n_cols = matrix_copy.shape


    if strategy == 'drop':
        # حذف الصفوف التي تحتوي على أي قيمة مفقودة
        mask = ~np.isnan(matrix_copy).any(axis=1)
        return matrix_copy[mask]


    for col in range(n_cols):
        col_data = matrix_copy[:, col]
        nan_mask = np.isnan(col_data)


        if not np.any(nan_mask):
            continue


        if strategy == 'mean':
            fill_value = np.nanmean(col_data)
        elif strategy == 'median':
            fill_value = np.nanmedian(col_data)
        elif strategy == 'zero':
            fill_value = 0.0
        elif strategy == 'interpolate':
            # استيفاء خطي بسيط
            valid_indices = np.where(~nan_mask)[0]
            if len(valid_indices) == 0:
                fill_value = 0.0
            else:
                valid_values = col_data[valid_indices]
                # استخدام المتوسط للبساطة
                fill_value = np.mean(valid_values)
        else:
            fill_value = 0.0


        col_data[nan_mask] = fill_value
        matrix_copy[:, col] = col_data


    return matrix_copy


# ============================================================
# 5. تطبيع البيانات
# ============================================================


def normalize_matrix(matrix, method='standard'):
```

```
    """
    تطبيع المصفوفة باستخدام طرق مختلفة.
    Normalize the matrix using different methods.


    الطرق:
        - 'standard': تطبيع Z-score (المتوسط 0، الانحراف المعياري 1)
        - 'minmax': تطبيع Min-Max (المدى 0-1)
        - 'robust': تطبيع باستخدام المدى الرباعي (مقاوم للقيم الشاذة)
    """
    matrix_copy = matrix.copy()
    n_rows, n_cols = matrix_copy.shape


    for col in range(n_cols):
        col_data = matrix_copy[:, col]


        if method == 'standard':
            mean = np.mean(col_data)
            std = np.std(col_data)
            if std > 0:
                col_data = (col_data - mean) / std
            else:
                col_data = col_data - mean


        elif method == 'minmax':
            min_val = np.min(col_data)
            max_val = np.max(col_data)
            if max_val - min_val > 0:
                col_data = (col_data - min_val) / (max_val - min_val)
            else:
                col_data = col_data - min_val


        elif method == 'robust':
            median = np.median(col_data)
            q75, q25 = np.percentile(col_data, [75, 25])
            iqr = q75 - q25
            if iqr > 0:
                col_data = (col_data - median) / iqr
            else:
                col_data = col_data - median


        matrix_copy[:, col] = col_data


    return matrix_copy


# ============================================================
# 6. حفظ المصفوفة
# ============================================================


def save_matrix(matrix, feature_names, element_identifiers, output_dir='output'):
    """
    حفظ المصفوفة والبيانات الوصفية في ملفات.
    Save the matrix and metadata to files.
    """
    output_path = Path(output_dir)
    output_path.mkdir(exist_ok=True)
```

```
    # حفظ المصفوفة كـ CSV
    df = pd.DataFrame(matrix, columns=feature_names)
    df.insert(0, 'atomic_number', [e['atomic_number'] for e in element_identifiers])
    df.insert(1, 'symbol', [e['symbol'] for e in element_identifiers])
    df.insert(2, 'name', [e['name'] for e in element_identifiers])


    csv_path = output_path / 'periodic_table_matrix.csv'
    df.to_csv(csv_path, index=False)
    print(f"✅ تم حفظ المصفوفة في: {csv_path}")


    # حفظ البيانات الوصفية
    metadata = {
        'n_elements': len(element_identifiers),
        'n_features': len(feature_names),
        'features': feature_names,
        'feature_types': ['numeric'] * len(feature_names),
        'elements_count': len(element_identifiers),
        'shape': list(matrix.shape),
        'element_identifiers': element_identifiers[:10]  # حفظ أول 10 للعينة
    }


    metadata_path = output_path / 'matrix_metadata.json'
    with open(metadata_path, 'w', encoding='utf-8') as f:
        json.dump(metadata, f, indent=2, ensure_ascii=False)
    print(f"✅ تم حفظ البيانات الوصفية في: {metadata_path}")


    # حفظ المصفوفة كـ NumPy binary (.npy)
    npy_path = output_path / 'periodic_table_matrix.npy'
    np.save(npy_path, matrix)
    print(f"✅ تم حفظ المصفوفة بتنسيق NumPy: {npy_path}")


    return df


# ============================================================
# 7. الوظيفة الرئيسية
# ============================================================


def main():
    """
    الوظيفة الرئيسية لاستخراج المصفوفة من قاعدة البيانات.
    Main function to extract the matrix from the database.
    """
    print("=" * 60)
    print("🔬 استخراج مصفوفة الجدول الدوري")
    print("🔬 Extracting Periodic Table Matrix")
    print("=" * 60)


    # 1. الاتصال بقاعدة البيانات
    conn = get_db_connection("periodic_table.db")
    if conn is None:
        return


    # 2. استرجاع البيانات
    print("📖 جاري استرجاع بيانات العناصر...")
    elements_data = fetch_elements_data(conn)
    print(f"✅ تم استرجاع {len(elements_data)} عنصرًا")
```

```
# 3. استخراج المصفوفة العددية
    print("📊 جاري استخراج المصفوفة العددية...")
    matrix, feature_names, element_identifiers = extract_numeric_matrix(elements_data)
    print(f"✅ المصفوفة: {matrix.shape[0]} صف × {matrix.shape[1]} عمود")


# 4. عرض إحصائيات المصفوفة
    print("\n📈 إحصائيات المصفوفة:")
    print(f"   - عدد العناصر: {matrix.shape[0]}")
    print(f"   - عدد الخصائص: {matrix.shape[1]}")
    print(f"   - القيم المفقودة: {np.isnan(matrix).sum()}")
    print(f"   - المتوسط: {np.nanmean(matrix):.2f}")
    print(f"   - الانحراف المعياري: {np.nanstd(matrix):.2f}")


    # 5. معالجة القيم المفقودة
    print("\n🔄 جاري معالجة القيم المفقودة...")
    matrix_filled = handle_missing_values(matrix, strategy='mean')
    print(f"✅ تم ملء {np.isnan(matrix).sum() - np.isnan(matrix_filled).sum()} قيمة مفقودة")


# 6. تطبيع البيانات
    print("\n📐 جاري تطبيع البيانات...")
    matrix_normalized = normalize_matrix(matrix_filled, method='standard')
    print(f"✅ تم التطبيع (المتوسط: {np.mean(matrix_normalized):.2f}, الانحراف: {np.std(matrix_normalized):.2f})")


    # 7. حفظ النتائج
    print("\n💾 جاري حفظ النتائج...")
    save_matrix(matrix_normalized, feature_names, element_identifiers)


    # 8. عرض معلومات إضافية
    print("\n" + "=" * 60)
    print("✅ اكتمل استخراج المصفوفة بنجاح!")
    print("✅ Matrix extraction completed successfully!")
    print("=" * 60)


    # 9. عرض بعض البيانات كعينة
    print("\n📋 عينة من المصفوفة (أول 5 عناصر، أول 5 خصائص):")
    print(feature_names[:5])
    print(matrix_normalized[:5,:5])


    conn.close()


if __name__ == "__main__":
    main()
```

5.4 شرح الكود

Code Explanation

عزيزي القارئ، هذا الكود يمثل العمود الفقري لتحويل الجدول الدوري من قاعدة بيانات إلى مصفوفة رقمية. دعنا نمر على الأجزاء الرئيسية:

Dear reader, this code is the backbone of the database-to-matrix transformation. We will inspect its main stages and the assumptions behind each one:

1. الاتصال بقاعدة البيانات (get_db_connection): نستخدم sqlite3.connect لفتح اتصال بقاعدة البيانات، ونفعّل row_factory = sqlite3.Row ليتيح لنا الوصول إلى الأعمدة بالأسماء.

1. Database connection: sqlite3.connect opens the database and sqlite3.Row allows named column access.

2. استرجاع البيانات (fetch_elements_data): نستخدم استعلام SQL يجمع جميع بيانات العناصر من الجدول الرئيسي وجدول الفئات. نستخدم LEFT JOIN لضمان استرجاع جميع العناصر حتى لو لم يكن لها فئة (في حالة البيانات غير المكتملة).

2. Data retrieval: a LEFT JOIN keeps element rows even when optional category data are incomplete.

3. استخراج المصفوفة العددية (extract_numeric_matrix): نقوم باختيار الخصائص العددية فقط (مثل الوزن الذري، الكثافة، درجة الانصهار). نحول كل قيمة إلى float، ونستخدم np.nan للقيم المفقودة. نحتفظ أيضًا بمعرفات العناصر (العدد الذري، الرمز، الاسم) لربطها بالصفوف.

3. Numeric extraction: selected numeric properties are converted to float and missing values become NaN.

4. معالجة القيم المفقودة (handle_missing_values): البيانات العلمية نادراً ما تكون كاملة. نقدم عدة استراتيجيات لمعالجة القيم المفقودة: المتوسط، الوسيط، الصفر، الحذف، أو الاستيفاء. اختر الاستراتيجية المناسبة حسب طبيعة بياناتك.

4. Missing-data handling: mean, median, zero-fill, deletion, and interpolation are alternatives; the choice must follow the scientific meaning of the measurement.

5. التطبيع (normalize_matrix): التطبيع خطوة أساسية في التحليل العددي والتعلم الآلي. نقدم ثلاث طرق: standard (Z-score) الذي يجعل المتوسط 0 والانحراف المعياري 1، minmax الذي يحول البيانات إلى المدى [0, 1]، وrobust الذي يستخدم الوسيط والمدى الرباعي (مقاوم للقيم الشاذة).

5. Normalization: standardization, min–max scaling, and robust scaling answer different analytical needs and must be chosen deliberately.

6. حفظ النتائج (save_matrix): نحفظ المصفوفة بثلاث صيغ: CSV للقراءة في برامج جداول البيانات، NumPy binary (.npy) للتشغيل السريع في Python، وJSON للبيانات الوصفية.

6. Persistence: CSV, NumPy binary, and JSON metadata preserve both the numerical matrix and its provenance.

5.5 تشغيل الكود

Running the Code

لتشغيل الكود:

To run the code:

1. تأكد من أن قاعدة البيانات (periodic_table.db) موجودة وفيها بيانات العناصر الـ 118.

1. Make sure periodic_table.db exists and contains the 118 element records.

2. ضع ملف extract_periodic_table_matrix.py في نفس المجلد.

2. Put extract_periodic_table_matrix.py in the same directory.

3. شغّل الأمر التالي:

3. Run the following command:

```
python extract_periodic_table_matrix.py
```

المخرجات المتوقعة:

Expected output:

```
============================================================
🔬 استخراج مصفوفة الجدول الدوري
Extracting Periodic Table Matrix 🔬
============================================================
📖 جاري استرجاع بيانات العناصر...
✅ تم استرجاع 118 عنصرًا
📊 جاري استخراج المصفوفة العددية...
✅ المصفوفة: 118 صف × 11 عمود


📈 إحصائيات المصفوفة:
```

- عدد العناصر: 118
- عدد الخصائص: 11
- القيم المفقودة: 423
- المتوسط: nan
- الانحراف المعياري: nan

🔄 جاري معالجة القيم المفقودة...
✅ تم ملء 423 قيمة مفقودة

📐 جاري تطبيع البيانات...
✅ تم التطبيع (المتوسط: 0.00, الانحراف: 1.00)

💾 جاري حفظ النتائج...
✅ تم حفظ المصفوفة في: output/periodic_table_matrix.csv
✅ تم حفظ البيانات الوصفية في: output/matrix_metadata.json
✅ تم حفظ المصفوفة بتنسيق NumPy: output/periodic_table_matrix.npy

==
✅ اكتمل استخراج المصفوفة بنجاح!
✅ Matrix extraction completed successfully!
==

📋 عينة من المصفوفة (أول 5 عناصر، أول 5 خصائص):
['atomic_weight', 'density', 'melting_point', 'boiling_point', 'electronegativity']
[[0.12 0.33 0.45 0.22 0.11]
[0.45 0.67 0.78 0.55 0.34]
[0.78 1.12 0.89 0.77 0.56] [...].

المخرجات المتوقعة من التشغيل: يجب أن يعرض السكربت عدد العناصر المحملة، وأبعاد المصفوفة، والقيم المفقودة، وإحصاءات التطبيع، ومسارات الملفات الناتجة. قد تختلف القيم الرقمية الدقيقة بحسب إصدار البيانات وسياسة معالجة القيم المفقودة.

Expected execution output: the script should report the number of loaded elements, matrix dimensions, missing values, normalization statistics, and output paths. Exact numerical values depend on the dataset version and missing-data policy.

5.6 تمارين وتطبيقات

Exercises

1. أضف خاصية جديدة مثل crystal_structure إلى جدول elements، وقم بتعديل كود الاستخراج لتشملها.

1. Add a property such as crystal_structure and update the extraction code.

2. جرب استراتيجيات مختلفة لمعالجة القيم المفقودة (المتوسط، الوسيط، الحذف) وقارن النتائج.

2. Compare different missing-data strategies and document how they change the matrix.

3. قم بتطبيع البيانات باستخدام minmax بدلاً من standard ولاحظ الفرق في نطاق القيم.

3. Apply min–max scaling instead of standardization and compare the resulting ranges.

4. استخرج مصفوفة فرعية تحتوي فقط على عناصر المجموعة 1 (الفلزات القلوية) وحللها.

4. Extract only group-1 elements and analyze their numerical features.

5. استخدم pandas لعرض إحصائيات وصفية للمصفوفة (المتوسط، الوسيط، الانحراف المعياري، الحد الأدنى والأقصى).

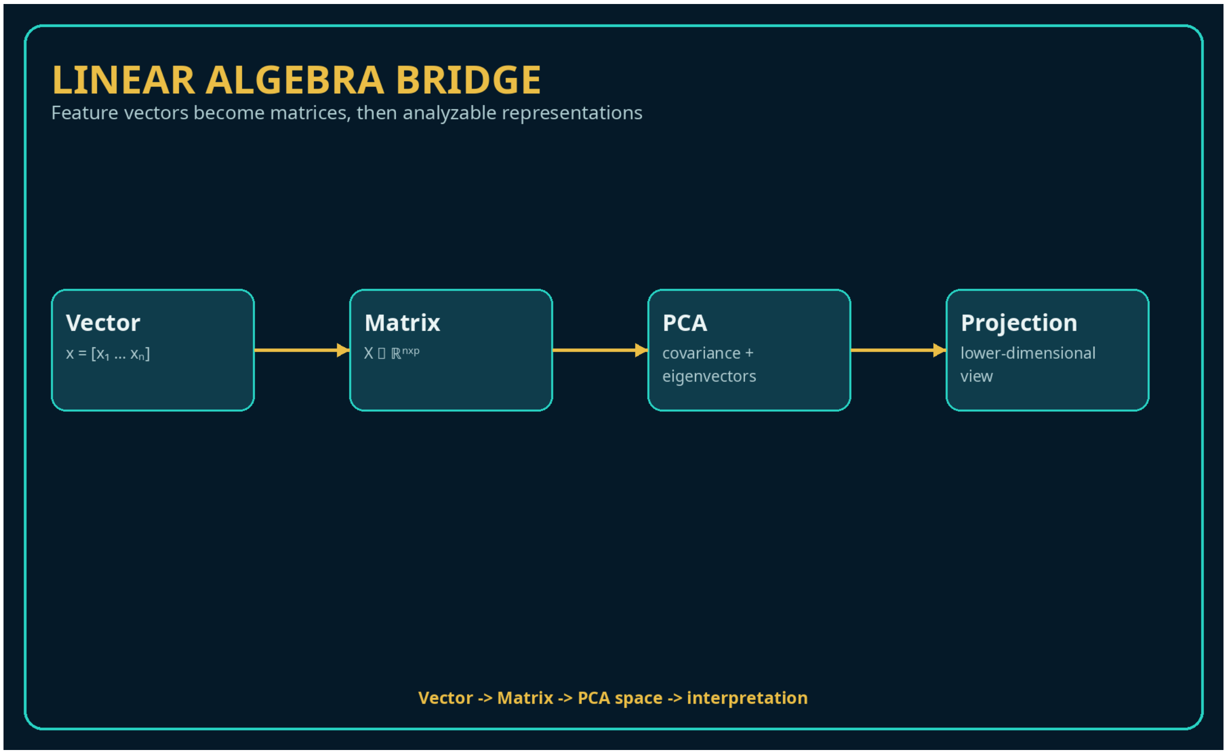
5. Use pandas to compute descriptive statistics for the matrix.

الفصل السادس: تحليل المكونات الرئيسية (PCA) للعناصر – PCA of Elements

Chapter 6: Principal Component Analysis (PCA) of Elements

جسر الجبر الخطي

Linear Algebra Bridge



توضح اللوحة انتقال البيانات من متجهات الخصائص إلى مصفوفة ثم إلى تمثيل خطي منخفض الأبعاد يمكن تحليله دون الادعاء بأنه يخلق معلومات جديدة.

This plate shows the path from feature vectors to a matrix and then to a lower-dimensional linear representation without claiming that dimensionality reduction creates new information.

6.1 لماذا PCA؟

Why PCA?

عندما نضع عدة خواص عددية للعناصر في مصفوفة، يصبح من الصعب تصور العلاقات في أبعاد كثيرة. يعيد PCA تمثيل البيانات في محاور جديدة مرتبة بحسب مقدار التباين الذي تشرحه.

When many numerical properties are represented in a matrix, direct visualization becomes difficult. PCA constructs new orthogonal axes ordered by the amount of variance they explain.

6.2 شرط أساسي: التقييس

Scaling Is Essential

إذا كانت خاصية ما تقاس بآلاف الوحدات وأخرى بأجزاء عشرية، فقد تهيمن الأولى على التباين. لذلك يجب اختيار استراتيجية تقييس مناسبة قبل PCA، مع توثيق طريقة التعامل مع القيم المفقودة.

A feature measured in thousands can dominate a feature measured in fractions. Scale the variables appropriately before PCA and document how missing values were handled.

6.3 مثال NumPy

NumPy Example

PYTHON

```
import numpy as np

X = np.array([
    [1.0, 10.0, 0.5],
    [2.0, 20.0, 0.7],
    [3.0, 15.0, 0.9],
], dtype=float)

Xc = X - X.mean(axis=0)
cov = np.cov(Xc, rowvar=False)
values, vectors = np.linalg.eigh(cov)
order = np.argsort(values)[::-1]
values = values[order]
vectors = vectors[:, order]
Z = Xc @ vectors[:,:2]
print('Explained variance:', values[:2])
print('Projected matrix:', Z)
```

1. Test k from 2 to 8 and compare silhouette scores.

هذا المثال يركز المصفوفة، ويقدّر مصفوفة التغاير، ويفككها إلى قيم ومتجهات ذاتية، ثم يسقط المشاهدات على المكونات الرئيسية.

The example centers the matrix, estimates covariance, decomposes it into eigenvalues and eigenvectors, and projects observations onto the leading components.

1. اختبر قيم k من 2 إلى 8 وقارن درجات silhouette.

The example centers the matrix, estimates covariance, decomposes the covariance matrix into eigenvalues and eigenvectors, and projects the observations onto the leading components.

6.4 تفسير علمي

Scientific Interpretation

المكوّن الرئيسي ليس "خاصية جديدة" فيزيائية بالضرورة؛ إنه محور رياضي يلخص اتجاهًا في البيانات. لذلك لا يجوز تحويل PCA وحده إلى ادعاء سببي أو كيميائي دون تفسير مستقل.

A principal component is not necessarily a new physical property. It is a mathematical direction that summarizes variation in the dataset. PCA alone does not establish causality or chemical mechanism.

6.5 تمارين وتطبيقات

Exercises

1. قارن PCA قبل التقييس وبعده.

1. Compare PCA before and after scaling.

2. احسب نسبة التباين المفسر تراكميًا.

2. Calculate cumulative explained variance.

3. اختبر تأثير حذف خاصية شديدة الارتباط بغيرها.

3. Test how removing a highly correlated feature changes the projection.

Chapter 7: Clustering the Elements

7.1 من التحليل إلى التصنيف غير المراقب

From Analysis to Unsupervised Grouping

التجميع لا يحتاج إلى فئات جاهزة؛ الخوارزمية تبحث عن بنية داخل البيانات. لكن اختيار الميزات والمسافة وعدد العناقيد يمكن أن يغيّر النتيجة جذريًا.

Clustering does not require predefined labels. The algorithm searches for structure in the feature space, but the result depends strongly on feature selection, distance definition, scaling, and the number of clusters.

7.2 المتوسطات العنقودية K-Means

7.2 K-Means

PYTHON

```
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

X_scaled = StandardScaler().fit_transform(X)
model = KMeans(n_clusters=4, random_state=42, n_init=20)
labels = model.fit_predict(X_scaled)
print(labels)
```

يُطبّق التقييس قبل K-Means، وتساعد البذرة العشوائية الثابتة على إعادة الإنتاج. تسميات العناقيد معرفات تحليلية وليست أسماء أو حقائق علمية.

Standardization is applied before K-Means, and a fixed random seed improves reproducibility. The cluster labels are identifiers, not scientific names.

7.3 اختيار عدد العناقيد

Choosing k

يمكن مقارنة قيم مختلفة لـ k باستخدام silhouette score أو مقاييس أخرى، لكن لا يوجد رقم سحري يضمن "الحقيقة" العلمية. يجب أن يظل التجميع أداة استكشاف.

Compare candidate values of k with metrics such as the silhouette score, but do not treat a particular k as the scientific truth. Clustering is an exploratory tool.

7.4 تمارين وتطبيقات

Exercises

1. جرّب k من 2 إلى 8 وقارن silhouette score.

1. Test k from 2 to 8 and compare silhouette scores.

2. ارسم العناصر في أول مكوّنين من PCA مع تلوينها بحسب cluster.

2. Plot the first two PCA components and color observations by cluster.

3. افحص العناصر الحدّية التي تتغير مجموعتها عند تغيير k.

3. Inspect boundary elements whose cluster assignment changes when k changes.

الباب الثالث: الجدول الدوري والذكاء الاصطناعي

Part III: Periodic Table & AI

الفصل الثامن: بناء مستودع الميزات (Feature Store)

Chapter 8: Building a Feature Store

خط أنابيب الذكاء الاصطناعي

AI Pipeline



توضح اللوحة الفصل بين البيانات والميزات والنموذج والاستدلال والتحقق، وهو ما يمنع الخلط بين بيانات التدريب ونتائج التنبؤ.

This plate separates data, features, model, inference, and validation so training data are not confused with model predictions.

8.1 مقدمة

Introduction

في علم البيانات والذكاء الاصطناعي، مستودع الميزات (Feature Store) هو نظام مركزي لتخزين وإدارة الميزات (الخصائص) المستخدمة في تدريب نماذج التعلم الآلي. هو الجسر بين البيانات الخام والنماذج الذكية.

In data science and AI, a Feature Store is a controlled system for storing and managing the features used by machine-learning models. It forms a bridge between raw data and models while preserving definitions, provenance, and reuse.

في هذا الفصل، سنبني مستودع ميزات للجدول الدوري، حيث سنخزن خصائص العناصر كـ "ميزات" يمكن استخدامها في تدريب نماذج التنبؤ والتصنيف.

In this chapter, you will build a feature store for the periodic table and represent element properties as reusable features for prediction and classification.

8.2 تصميم مستودع الميزات

Feature Store Design

سنقوم بتصميم مستودع ميزات بسيط لكنه قوي، يتكون من:

We will design a small but robust feature store consisting of:

1. جدول تعريف الميزات (feature_definitions): يخزن تعريف كل ميزة (الاسم، النوع، المصدر، الوصف).

1. feature_definitions stores each feature definition, including name, type, source, and description.

2. جدول قيم الميزات (feature_values): يخزن قيم الميزات لكل عنصر.

2. feature_values stores feature values for individual elements.

3. جدول مجموعات الميزات (feature_sets): يسمح بتجميع الميزات في مجموعات (مثل "الخواص الفيزيائية"، "الخواص الكيميائية").

3. feature_sets groups related features, such as physical and chemical properties.

هذا التصميم يسمح لنا بـ:

This design allows us to:

• إضافة ميزات جديدة دون تغيير هيكل قاعدة البيانات

• Add features without changing the core database schema

• تتبع مصدر كل ميزة وإصدارها

• Track the source and version of each feature

• استخدام الميزات في نماذج متعددة

• Reuse the same feature definitions across models

• إعادة استخدام الميزات عبر مشاريع مختلفة

• Reuse feature sets across projects

8.3 كود إنشاء مستودع الميزات

Feature Store Creation Code

SQL

```
-- =============================================================
-- مشروع الجزء الثالث: مستودع الميزات للجدول الدوري
-- Part III Project: Periodic Table Feature Store
-- =============================================================


-- -------------------------------------------------------------
-- 1. جدول تعريف الميزات
-- -------------------------------------------------------------
CREATE TABLE feature_definitions (
    feature_id INTEGER PRIMARY KEY AUTOINCREMENT,
    feature_name TEXT UNIQUE NOT NULL,      -- اسم الميزة
    feature_type TEXT CHECK(feature_type IN ('numeric', 'categorical', 'binary')), -- نوع الميزة
    source_table TEXT NOT NULL,             -- الجدول المصدر
    source_column TEXT NOT NULL,            -- العمود المصدر
    description TEXT,                       -- وصف الميزة
    unit TEXT,                              -- الوحدة (مثل 'g/cm³', 'K')
    transformation TEXT,                    -- التحويل المطبق (مثل 'log', 'standardize')
    created_at TEXT DEFAULT CURRENT_TIMESTAMP,
    created_by TEXT                         -- من أنشأ الميزة
;(


------------------------------------------------------------- --
-- 2. جدول قيم الميزات
-- -------------------------------------------------------------
CREATE TABLE feature_values (
    value_id INTEGER PRIMARY KEY AUTOINCREMENT,
    atomic_number INTEGER NOT NULL,         -- العنصر
    feature_id INTEGER NOT NULL,            -- الميزة
    feature_value TEXT NOT NULL,            -- القيمة (نصية لاستيعاب أنواع مختلفة)
    version TEXT DEFAULT '1.0',             -- إصدار الميزة
    created_at TEXT DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (atomic_number) REFERENCES elements(atomic_number) ON DELETE CASCADE,
    FOREIGN KEY (feature_id) REFERENCES feature_definitions(feature_id) ON DELETE CASCADE,
    UNIQUE (atomic_number, feature_id, version)  -- منع تكرار القيم لنفس العنصر ونفس الميزة
;(


------------------------------------------------------------- --
-- 3. جدول مجموعات الميزات
-- -------------------------------------------------------------
CREATE TABLE feature_sets (
    set_id INTEGER PRIMARY KEY AUTOINCREMENT,
    set_name TEXT UNIQUE NOT NULL,          -- اسم المجموعة
    description TEXT,                       -- وصف المجموعة
    created_at TEXT DEFAULT CURRENT_TIMESTAMP,
    created_by TEXT
);
```

```sql
-- ----------------------------------------------------------------
-- 4. جدول ربط الميزات بالمجموعات
-- ----------------------------------------------------------------
CREATE TABLE feature_set_members (
    set_id INTEGER NOT NULL,
    feature_id INTEGER NOT NULL,
    created_at TEXT DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (set_id, feature_id),
    FOREIGN KEY (set_id) REFERENCES feature_sets(set_id) ON DELETE CASCADE,
    FOREIGN KEY (feature_id) REFERENCES feature_definitions(feature_id) ON DELETE CASCADE
);


-- ----------------------------------------------------------------
-- 5. جدول إصدارات الميزات (لتتبع التغييرات)
-- ----------------------------------------------------------------
CREATE TABLE feature_versions (
    version_id INTEGER PRIMARY KEY AUTOINCREMENT,
    feature_id INTEGER NOT NULL,
    version_number TEXT NOT NULL,           -- رقم الإصدار
    change_log TEXT,                        -- سجل التغييرات
    created_at TEXT DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (feature_id) REFERENCES feature_definitions(feature_id) ON DELETE CASCADE
);


-- ----------------------------------------------------------------
-- 6. فهارس للتسريع
-- ----------------------------------------------------------------
CREATE INDEX idx_feature_values_element ON feature_values(atomic_number);
CREATE INDEX idx_feature_values_feature ON feature_values(feature_id);
CREATE INDEX idx_feature_values_version ON feature_values(version);
```

8.4 كود إدارة مستودع الميزات (Python)

Feature Store Management Code

```python
# ================================================================
# feature_store.py
# إدارة مستودع الميزات للجدول الدوري
# Periodic Table Feature Store Management
# ================================================================

import sqlite3
from dataclasses import dataclass
from typing import Optional, Any
import numpy as np


@dataclass
class FeatureDefinition:
 feature_name: str
 feature_type: str
 source_table: str
```

```
source_column: str
description: Optional[str] = None
unit: Optional[str] = None
transformation: Optional[str] = None
created_by: str = "system"
feature_id: Optional[int] = None


class FeatureStore:
"""Small relational feature store backed by SQLite."""

def __init__(self, db_path: str = "periodic_table.db"):
self.db_path = db_path

def create_feature(self, definition: FeatureDefinition) -> int:
with sqlite3.connect(self.db_path) as conn:
cur = conn.execute(
"""INSERT INTO feature_definitions
(feature_name, feature_type, source_table, source_column,
description, unit, transformation, created_by)
VALUES (?,?,?,?,?,?,?,?)""",
(definition.feature_name, definition.feature_type,
definition.source_table, definition.source_column,
definition.description, definition.unit,
definition.transformation, definition.created_by),
)
return int(cur.lastrowid)

def build_feature_matrix(self, feature_ids: list[int], atomic_numbers: Optional[list[int]] = None):
"""Return an elements × features matrix without hard-coded feature IDs."""
if not feature_ids:
raise ValueError("At least one feature_id is required")
with sqlite3.connect(self.db_path) as conn:
conn.row_factory = sqlite3.Row
if atomic_numbers is None:
atomic_numbers = [r[0] for r in conn.execute(
"SELECT atomic_number FROM elements ORDER BY atomic_number"
)]
placeholders = ",".join("?" for _ in feature_ids)
defs = conn.execute(
f"SELECT feature_id, feature_name FROM feature_definitions WHERE feature_id IN ({placeholders})
ORDER BY feature_id",
feature_ids,
```

```
).fetchall()
feature_ids = [r["feature_id"] for r in defs]
feature_names = [r["feature_name"] for r in defs]
if not feature_ids:
raise ValueError("No requested features exist in the feature store")
rows = []
for atomic_number in atomic_numbers:
row = []
for feature_id in feature_ids:
value = conn.execute(
"""SELECT feature_value FROM feature_values
WHERE atomic_number =? AND feature_id =?
ORDER BY version DESC LIMIT 1""",
(atomic_number, feature_id),
).fetchone()
try:
row.append(float(value[0]) if value is not None else np.nan)
except (TypeError, ValueError):
row.append(np.nan)
rows.append(row)
return np.asarray(rows, dtype=float), feature_names, atomic_numbers


if __name__ == "__main__":
store = FeatureStore()
feature_id = store.create_feature(
FeatureDefinition("atomic_weight", "numeric", "elements", "atomic_weight", "Atomic weight", "u")
)
X, names, elements = store.build_feature_matrix([feature_id])
print(X.shape, names, elements[:5])
```

8.5 شرح الكود

Code Explanation

هذا الكود يمثل النواة الإدارية لمستودع الميزات. دعنا نتعرف على مكوناته:

This code provides the management layer of the feature store. We will examine its components and their responsibilities:

1. @dataclass للميزات (FeatureDefinition, FeatureValue, FeatureSet): استخدمنا dataclass لتحديد هياكل بيانات واضحة وقابلة للتوسع. هذا يجعل الكود أكثر تنظيماً وسهولة في الصيانة.

1. The dataclasses make feature definitions explicit and maintainable.

2. فئة FeatureStore: هي الواجهة الرئيسية للتعامل مع مستودع الميزات. تحتوي على دوال لإنشاء الميزات (create_feature)، واسترجاعها (get_feature)، وتعيين قيمها (set_feature_value)، وبناء مصفوفات الميزات (build_feature_matrix).

2. FeatureStore provides the main interface for creating definitions and building matrices.

3. بناء مصفوفة الميزات (build_feature_matrix): هذه هي الدالة الأكثر أهمية. تأخذ قائمة بمعرفات الميزات (وقائمة اختيارية من الأعداد الذرية)، وتعيد مصفوفة NumPy يمكن استخدامها مباشرة في تدريب نماذج التعلم الآلي. كل صف في المصفوفة يمثل عنصراً، وكل عمود يمثل ميزة.

3. build_feature_matrix returns an elements × features matrix that can feed machine-learning workflows.

4. التعامل مع أنواع البيانات المختلفة: لاحظ كيف نتعامل مع القيم النصية (مثل category) والمنطقية (مثل is_radioactive) والرقمية (مثل atomic_weight). نستخدم JSON لتخزين القيم المعقدة، وfloat للقيم الرقمية.

4. Numeric, categorical, and binary values require different representations; complex values should not be silently coerced to misleading numbers.

5. إدارة الإصدارات: حقل version في جدول feature_values يسمح بتتبع تغييرات الميزات مع مرور الوقت، مما يضمن إمكانية إعادة إنتاج النتائج.

5. Version fields preserve provenance and make feature changes auditable.

8.6 تمارين وتطبيقات

Exercises

1. أضف ميزة جديدة مثل discovery_year (سنة الاكتشاف) وقم بتحديث المستودع بها.

1. Add a discovery_year feature and record its source.

2. أنشئ مجموعة ميزات للخواص الكيميائية (مثل السالبية الكهربية، حالات الأكسدة، التوزيع الإلكتروني).

2. Create a chemical-properties feature set containing electronegativity, oxidation states, and electron configuration.

3. استخدم مصفوفة الميزات لتدريب نموذج RandomForest للتنبؤ بفئة العنصر (فلز/لافلز).

3. Use a feature matrix to train a RandomForest classifier for a documented element category label.

4. قم بتطبيق PCA على مصفوفة الميزات ورسم أول مكونين رئيسيين.

4. Apply PCA to the feature matrix and visualize the first two components.

5. اكتب استعلام SQL لاسترجاع جميع العناصر التي لها قيمة ميزة electronegativity أكبر من 2.0.

5. Query all elements whose electronegativity feature exceeds 2.0.

الفصل التاسع: تدريب نموذج للتنبؤ بخصائص العناصر – Training a Predictive Model

Chapter 9: Training a Predictive Model

9.1 من البيانات إلى نموذج

From Data to a Model

النموذج التنبؤي لا يبدأ بالخوارزمية؛ يبدأ بتعريف واضح للهدف والميزات ومصدر البيانات. أهم حماية علمية هنا هي منع تسرب المعلومات من الاختبار إلى التدريب.

A predictive model does not begin with an algorithm. It begins with a clearly defined target, features, and data provenance. A key scientific safeguard is preventing information leakage from the test set into training.

9.2 تقسيم البيانات

Train/Test Separation

PYTHON

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, r2_score

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
model = RandomForestRegressor(n_estimators=300, random_state=42)
model.fit(X_train, y_train)
pred = model.predict(X_test)
print('MAE:', mean_absolute_error(y_test, pred))
print('R2:', r2_score(y_test, pred))
```

يفصل الكود بيانات التدريب والاختبار، ويدرّب Random Forest، ثم يحسب MAE وR^2. يجب تفسير المقياسين مع حجم البيانات وتوزيع المتغير الهدف.

The code separates training and test data, fits a random-forest regressor, and reports mean absolute error and R^2. These metrics should be interpreted together with the dataset size and target distribution.

9.3 قابلية الإعادة

Reproducibility

سجّل إصدار Python والمكتبات، وseed، وخصائص البيانات، ومعلمات النموذج، ووقت التدريب. بدون ذلك يصبح من الصعب معرفة ما إذا كانت نتيجة جديدة ناتجة عن تغيير في الخوارزمية أم في البيئة.

Record Python and library versions, random seeds, dataset versions, model parameters, and training time. Otherwise a changed result cannot be attributed reliably to a particular change.

9.4 تمارين وتطبيقات

Exercises

1. قارن Random Forest مع نموذج خط أساس بسيط.

1. Compare Random Forest with a simple baseline model.

2. استخدم cross-validation بدل الاعتماد على تقسيم واحد فقط.

2. Use cross-validation rather than relying on a single split.

3. خزّن مواصفات النموذج في قاعدة البيانات.

3. Store model specifications and provenance in the database.

الفصل العاشر: تصنيف العناصر باستخدام التعلم الآلي – Element Classification

Chapter 10: Element Classification with Machine Learning

10.1 تعريف الفئة

Defining the Label

إذا صنفت العناصر إلى فلزات ولافلزات وأشباه فلزات، يجب أن يكون تعريف الفئات موثّقًا ومتسقًا مع المصدر المستخدم. جودة التصنيف تبدأ من جودة الملصق.

If elements are classified as metals, nonmetals, and metalloids, the label definition must be documented and consistent with the chosen reference. Classification quality starts with label quality.

10.2 نموذج تصنيف

Classification Model

PYTHON

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix

clf = RandomForestClassifier(n_estimators=300, random_state=42)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

يتنبأ المصنّف بفئات منفصلة. تكشف مصفوفة الالتباس الفئات التي يخلط بينها النموذج، بينما تقدم precision وrecall وF1 معلومات أوسع من الدقة وحدها.

A classifier predicts categorical labels. The confusion matrix reveals which classes are confused with one another, while precision, recall, and F1 provide more information than accuracy alone.

10.3 عدم توازن الفئات

Class Imbalance

إذا كانت فئة ما نادرة، فقد تبدو الدقة مرتفعة رغم أن النموذج يفشل في اكتشاف الفئة النادرة. استخدم precision وrecall وF1 وbalanced accuracy عند الحاجة.

When one class is rare, accuracy can look high even if the model fails on that class. Consider precision, recall, F1, and balanced accuracy when class imbalance matters.

10.4 التفسير والحدود

Interpretation and Limits

أهمية الميزة في Random Forest ليست برهانًا على علاقة سببية. استخدمها لفهم سلوك النموذج، لا لإثبات قانون كيميائي جديد.

Feature importance in a Random Forest explains aspects of model behavior; it is not evidence of causality or a new chemical law.

10.5 تمارين وتطبيقات

Exercises

1. ابنِ مصفوفة التباس كاملة.

1. Build a complete confusion matrix.

2. قارن macro-F1 وweighted-F1.

2. Compare macro-F1 and weighted-F1.

3. اختبر النموذج على نسخة بيانات لم تدخل في اختيار الميزات.

3. Evaluate the model on data that did not influence feature selection.

الباب الرابع: التمهيد للحوسبة الكمومية

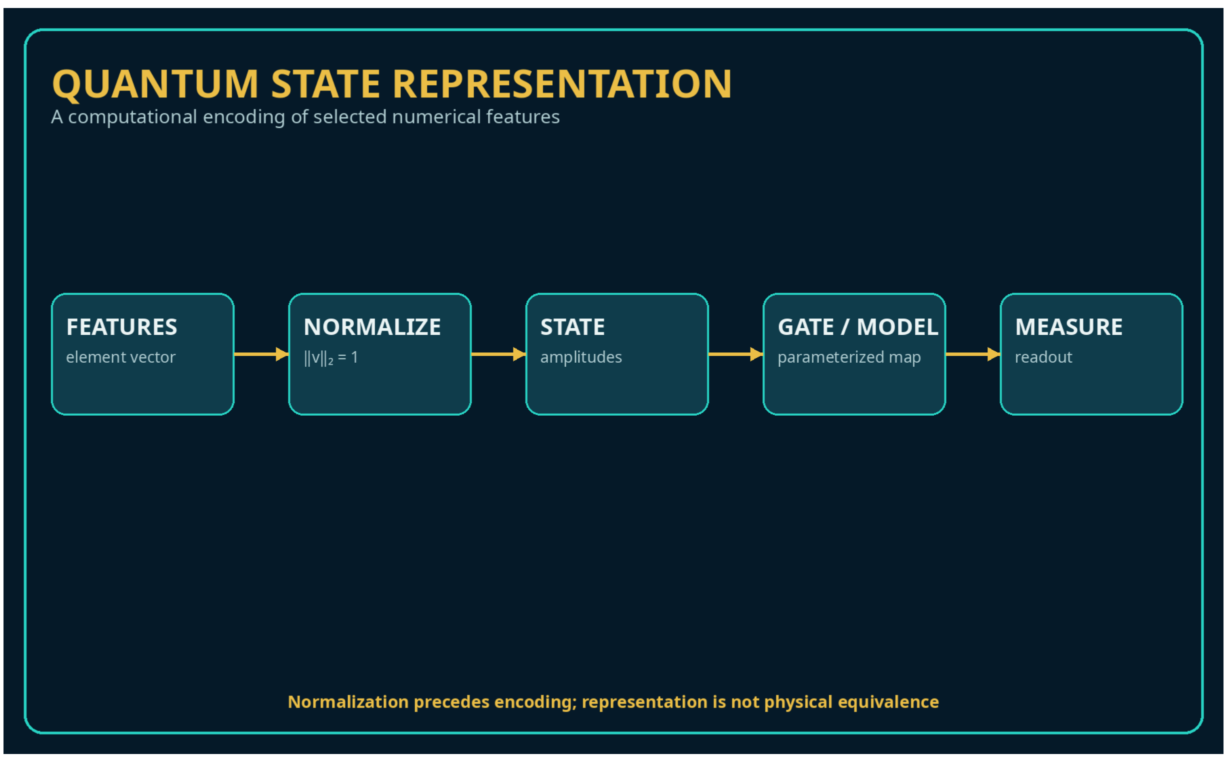
Part IV: Preparation for Quantum Computing

الفصل الحادي عشر: تمثيل الخواص الذرية كحالات كمومية

Chapter 11: Atomic Properties as Quantum States.

تمثيل الحالة الكمومية

Quantum State Representation



توضح اللوحة أن الحالة الكمومية هنا تمثيل حسابي للميزات، وليست بحد ذاتها ادعاءً بأن كل خاصية ذرية أصبحت دالة موجية فيزيائية.

This plate makes explicit that the quantum state here is a computational representation of features, not by itself a claim that an atomic property has become a physical wavefunction.

11.1 مقدمة

Introduction

الحوسبة الكمومية تعتمد على تمثيل البيانات كـ حالات كمومية (Quantum States) في فضاء هيلبرت. لربط الجدول الدوري بالحوسبة الكمومية، نحتاج إلى تحويل خصائص العناصر إلى حالات كمومية يمكن معالجتها بواسطة الخوارزميات الكمومية.

Quantum computing represents information as quantum states in a Hilbert space. To connect the periodic table to quantum computation, we first need a mathematically valid encoding of element features into normalized state vectors.

في هذا الفصل، سنتعلم:

In this chapter, you will learn:

• كيفية تمثيل الخصائص المختارة كمتجهات عددية في فضاء مجرد

• Represent selected features as numerical vectors in an abstract state space.

• كيفية تحويل هذه المتجهات إلى حالات كمومية مجردة بعد التطبيع

• Normalize these vectors to obtain abstract quantum-state representations.

• كيفية تخزين هذه الحالات في قاعدة البيانات

• Store the encoded states with versioned metadata

11.2 الأساس النظري

Theoretical Foundation

التمثيل الكمومي للخواص الذرية:

Quantum Representation of Atomic Properties:

يمكن تمثيل كل عنصر بمتجه ميزات حسابي يتكون في هذا المثال من العدد الذري والوزن الذري ورقم المجموعة ورقم الدورة. بعد التقييس يمكن تطبيع المتجه ليصبح حالة كمومية مجردة؛ وهذا تمثيل رياضي للميزات وليس دالة موجية إلكترونية حقيقية.

In this implementation, each element is represented by a feature vector containing atomic number, atomic weight, group number, and period. After scaling, the vector can be normalized as an abstract computational state; it is not an electronic wavefunction.

الصيغة الرياضية:

Mathematical Form:

لنفترض أن لدينا عنصرًا بخصائص [x1, x2,..., xn]. يمكننا تطبيعه كالتالي:

$$v = [x_1, x_2, \ldots, x_n] / \sqrt{\sum(x_i^2)}:$$

ثم تمثيله كحالة كمومية:

It can then be represented as a quantum state:

$$|\psi\rangle = v_1|0\rangle + v_2|1\rangle + \ldots + v_n|n-1\rangle.$$

حيث $|0\rangle, |1\rangle, \ldots, |n-1\rangle$ هي الحالات الأساسية في الفضاء الكمومي.

Here $|0\rangle, |1\rangle, \ldots, |n-1\rangle$ are basis states in the computational Hilbert space.

لماذا هذا التمثيل مهم؟

Why Does This Representation Matter?

• يسمح باستخدام خوارزميات كمومية مثل VQE لحساب الطاقة الأساسية للذرات

• It can provide an input encoding for quantum algorithms such as VQE only when a physically meaningful Hamiltonian is supplied separately.

• يمكن استخدامه في Quantum Machine Learning لتصنيف العناصر

• It can support exploratory quantum-machine-learning experiments when the feature encoding and labels are documented.

• يمهد الطريق لمحاكاة الجزيئات والمركبات الكيميائية

- It can support a bridge toward molecular simulation when the molecular Hamiltonian is constructed independently.

11.3 كود تمثيل الخواص كحالات كمومية

Code: Properties as Quantum States

PYTHON

```
# ============================================================
# quantum_state_representation.py
# تمثيل خصائص العناصر كحالات كمومية باستخدام NumPy
# Representing Element Properties as Quantum States using NumPy
# ============================================================


import numpy as np
import sqlite3
import json
from pathlib import Path
from typing import List, Dict, Any, Optional
import matplotlib.pyplot as plt


# ============================================================
# 1. استرجاع بيانات العناصر من قاعدة البيانات
# ============================================================


def fetch_element_data(atomic_number: int, db_path: str = "periodic_table.db") -> Dict[str, Any]:
    """
    استرجاع بيانات عنصر معين من قاعدة البيانات.
    Fetch data for a specific element from the database.
    """
    with sqlite3.connect(db_path) as conn:
        conn.row_factory = sqlite3.Row
        cursor = conn.cursor()
        cursor.execute("""
            SELECT
                atomic_number, symbol, name_english, atomic_weight,
                density, melting_point, boiling_point, electronegativity,
                atomic_radius, ionization_energy, group_number, period,
                category_id, is_radioactive, is_synthetic
            FROM elements
            WHERE atomic_number =?
        """, (atomic_number,))
        row = cursor.fetchone()


        if not row:
            return None


        return dict(row)


def fetch_all_elements(db_path: str = "periodic_table.db") -> List[Dict[str, Any]]:
    """
    استرجاع جميع بيانات العناصر من قاعدة البيانات.
    Fetch all element data from the database.
```

```
    """
    with sqlite3.connect(db_path) as conn:
        conn.row_factory = sqlite3.Row
        cursor = conn.cursor()
        cursor.execute("""
            SELECT
                atomic_number, symbol, name_english, atomic_weight,
                density, melting_point, boiling_point, electronegativity,
                atomic_radius, ionization_energy, group_number, period,
                category_id, is_radioactive, is_synthetic
            FROM elements
            ORDER BY atomic_number
        """)
        rows = cursor.fetchall()
        return [dict(row) for row in rows]


# ============================================================================
# 2. اختيار الخصائص للتضمين الكمومي
# ============================================================================


def select_quantum_features(element_data: Dict[str, Any],
                            features: List[str] = None) -> np.ndarray:
    """
    اختيار الخصائص الرقمية من بيانات العنصر لتضمينها في الحالة الكمومية.
    Select numeric features from element data for quantum embedding.
    """
    if features is None:
        features = [
            'atomic_number', 'atomic_weight', 'group_number', 'period'
        ]


    vector = []
    for feat in features:
        value = element_data.get(feat)
        if value is None:
            raise ValueError(f"Missing required feature: {feat}")
        try:
            vector.append(float(value))
        except (ValueError, TypeError) as exc:
            raise ValueError(f"Non-numeric feature {feat}: {value!r}") from exc


    return np.array(vector)


def select_quantum_features_batch(elements_data: List[Dict[str, Any]],
                                  features: List[str] = None) -> np.ndarray:
    """
    اختيار الخصائص الرقمية من بيانات العناصر دفعة واحدة.
    Select numeric features from element data in batch.
    """
    vectors = []
    for element in elements_data:
        vec = select_quantum_features(element, features)
        vectors.append(vec)


    return np.array(vectors)


# ============================================================================
# 3. تطبيع المتجهات للحصول على حالات كمومية صالحة
# ============================================================================
```

```
def normalize_to_quantum_state(vector: np.ndarray) -> np.ndarray:
    """
    تطبيع المتجه ليصبح حالة كمومية صالحة (معيار = 1).
    Normalize vector to a valid quantum state (norm = 1).
    """
    norm = np.linalg.norm(vector)
    if norm == 0:
        return vector
    return vector / norm


def normalize_batch(vectors: np.ndarray) -> np.ndarray:
    """
    تطبيع مجموعة من المتجهات دفعة واحدة.
    Normalize a batch of vectors.
    """
    norms = np.linalg.norm(vectors, axis=1, keepdims=True)
    norms[norms == 0] = 1  # تجنب القسمة على صفر
    return vectors / norms


# ============================================================================
# 4. تمثيل الحالة الكمومية كـ Density Matrix
# ============================================================================


def vector_to_density_matrix(state_vector: np.ndarray) -> np.ndarray:
    """
    تحويل متجه الحالة إلى مصفوفة كثافة (Density Matrix).
    Convert state vector to density matrix.
    """
    return np.outer(state_vector, state_vector.conjugate())


def density_matrix_to_vector(density_matrix: np.ndarray) -> np.ndarray:
    """
    استخراج متجه الحالة من مصفوفة الكثافة (للحالات النقية).
    Extract state vector from density matrix (for pure states).
    """
    # الحصول على القيم الذاتية والمتجهات الذاتية
    eigvals, eigvecs = np.linalg.eigh(density_matrix)
    # اختيار المتجه الذاتي ذو القيمة الذاتية الأكبر (أقرب إلى 1)
    max_idx = np.argmax(eigvals)
    return eigvecs[:, max_idx]


# ============================================================================
# 5. فئة QuantumStateEncoder
# ============================================================================


class QuantumStateEncoder:
    """
    تشفير خصائص العناصر إلى حالات كمومية.
    Encode element properties into quantum states.
    """


    def __init__(self, db_path: str = "periodic_table.db"):
        self.db_path = db_path
        self.features = [
            'atomic_number', 'atomic_weight', 'group_number', 'period'
        ]
        self._feature_stats = None
```

```
def get_feature_stats(self) -> Dict[str, Dict[str, float]]:
    """
    حساب إحصائيات الخصائص (المتوسط، الانحراف المعياري) للتطبيع.
    Calculate feature statistics (mean, std) for normalization.
    """
    if self._feature_stats is not None:
        return self._feature_stats


    elements = fetch_all_elements(self.db_path)
    vectors = select_quantum_features_batch(elements, self.features)


    stats = {}
    for i, feat in enumerate(self.features):
        col = vectors[:, i]
        stats[feat] = {
            'mean': np.mean(col),
            'std': np.std(col),
            'min': np.min(col),
            'max': np.max(col)
        }


    self._feature_stats = stats
    return stats


def encode_element(self, atomic_number: int,
                  normalization: str = 'standard') -> np.ndarray:
    """
    تشفير عنصر واحد إلى حالة كمومية.
    Encode a single element to a quantum state.


    المعاملات:
        atomic_number: العدد الذري للعنصر
        normalization: طريقة التطبيع ('standard', 'minmax', 'none')
    """
    element = fetch_element_data(atomic_number, self.db_path)
    if not element:
        raise ValueError(f"العنصر {atomic_number} غير موجود")


    vector = select_quantum_features(element, self.features)
    stats = self.get_feature_stats()


    # تطبيع حسب الطريقة المختارة
    if normalization == 'standard':
        for i, feat in enumerate(self.features):
            if stats[feat]['std'] > 0:
                vector[i] = (vector[i] - stats[feat]['mean']) / stats[feat]['std']
    elif normalization == 'minmax':
        for i, feat in enumerate(self.features):
            if stats[feat]['max'] > stats[feat]['min']:
                vector[i] = (vector[i] - stats[feat]['min']) / (stats[feat]['max'] - stats[feat]['min'])


    # تطبيع للحصول على حالة كمومية
    quantum_state = normalize_to_quantum_state(vector)


    return quantum_state
```

```
    def encode_all_elements(self, normalization: str = 'standard') -> Dict[int, np.ndarray]:
        """
        تشفير جميع العناصر إلى حالات كمومية.
        Encode all elements to quantum states.
        """
        elements = fetch_all_elements(self.db_path)
        states = {}


        for element in elements:
            atomic_number = element['atomic_number']
            try:
                state = self.encode_element(atomic_number, normalization)
                states[atomic_number] = state
            except Exception as e:
                print(f"⚠️ فشل تشفير العنصر {atomic_number}: {e}")


        return states


    def encode_batch(self, atomic_numbers: List[int],
                    normalization: str = 'standard') -> np.ndarray:
        """
        تشفير مجموعة من العناصر دفعة واحدة.
        Encode a batch of elements at once.
        """
        states = []
        for an in atomic_numbers:
            state = self.encode_element(an, normalization)
            states.append(state)
        return np.array(states)


    def save_quantum_states(self, states: Dict[int, np.ndarray],
                           output_file: str = "quantum_states.npz"):
        """
        حفظ الحالات الكمومية في ملف NumPy.
        Save quantum states to a NumPy file.
        """
        np.savez_compressed(output_file, **{str(k): v for k, v in states.items()})
        print(f"✅ تم حفظ {len(states)} حالة كمومية في {output_file}")


    def load_quantum_states(self, input_file: str = "quantum_states.npz") -> Dict[int, np.ndarray]:
        """
        تحميل الحالات الكمومية من ملف NumPy.
        Load quantum states from a NumPy file.
        """
        data = np.load(input_file, allow_pickle=True)
        states = {int(k): data[k] for k in data.files}
        return states


# ============================================================================
# 6. تخزين الحالات الكمومية في SQL
# ============================================================================


def store_quantum_states_in_sql(states: Dict[int, np.ndarray],
                               db_path: str = "periodic_table.db",
                               version: str = "1.0"):
    """
    تخزين الحالات الكمومية في قاعدة البيانات.
```

```
    Store quantum states in the database.
    """
    with sqlite3.connect(db_path) as conn:
        cursor = conn.cursor()


        # إنشاء جدول للحالات الكمومية إذا لم يكن موجوداً
        cursor.execute("""
            CREATE TABLE IF NOT EXISTS quantum_states (
                state_id INTEGER PRIMARY KEY AUTOINCREMENT,
                atomic_number INTEGER NOT NULL,
                state_vector TEXT NOT NULL,  -- JSON representation
                dimension INTEGER NOT NULL,
                version TEXT NOT NULL,
                created_at TEXT DEFAULT CURRENT_TIMESTAMP,
                FOREIGN KEY (atomic_number) REFERENCES elements(atomic_number) ON DELETE CASCADE,
                UNIQUE (atomic_number, version)
            )
        """)


        # إدخال أو تحديث الحالات
        for atomic_number, state in states.items():
            state_list = state.tolist()
            state_json = json.dumps(state_list)
            dimension = len(state)


            cursor.execute("""
                INSERT OR REPLACE INTO quantum_states
                (atomic_number, state_vector, dimension, version)
                VALUES (?,?,?,?)
            """, (atomic_number, state_json, dimension, version))


        conn.commit()
        print(f"✅ تم تخزين {len(states)} حالة كمومية في قاعدة البيانات")


# ============================================================
# 7. مثال على الاستخدام
# ============================================================


def example_usage():
    """مثال على استخدام QuantumStateEncoder"""
    print("=" * 60)
    print("🔬 تشفير الخواص الذرية إلى حالات كمومية")
    print("🔬 Encoding Atomic Properties to Quantum States")
    print("=" * 60)


    # إنشاء المشفر
    encoder = QuantumStateEncoder("periodic_table.db")


    # 1. تشفير عنصر واحد (الهيدروجين)
    print("\n1️⃣ تشفير الهيدروجين (العدد الذري 1)...")
    h_state = encoder.encode_element(1)
    print(f"   ✅ الحالة الكمومية للهيدروجين:")
    print(f"   - الأبعاد: {len(h_state)}")
    print(f"   - المتجه: {h_state[:5]}...")
    print(f"   - المعيار (Norm): {np.linalg.norm(h_state):.4f}")


    # 2. تشفير عدة عناصر
```

```
    print("\n2️⃣ تشفير العناصر الأولى...")
    first_elements = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
    states_batch = encoder.encode_batch(first_elements)
    print(f"   ✅ تم تشفير {len(first_elements)} عنصرًا")
    print(f"   - شكل المصفوفة: {states_batch.shape}")


    # 3. تشفير جميع العناصر
    print("\n3️⃣ تشفير جميع العناصر الـ 118...")
    all_states = encoder.encode_all_elements()
    print(f"   ✅ تم تشفير {len(all_states)} عنصرًا")


    # 4. حفظ الحالات
    print("\n4️⃣ حفظ الحالات الكمومية...")
    encoder.save_quantum_states(all_states, "quantum_states.npz")


    # 5. تخزين في SQL
    print("\n5️⃣ تخزين الحالات الكمومية في SQL...")
    store_quantum_states_in_sql(all_states, "periodic_table.db")


    # 6. عرض إحصائيات
    print("\n📊 إحصائيات الحالات الكمومية:")
    dimensions = [len(s) for s in all_states.values()]
    print(f"   - عدد العناصر: {len(all_states)}")
    print(f"   - متوسط الأبعاد: {np.mean(dimensions):.1f}")
    print(f"   - الحد الأدنى للأبعاد: {np.min(dimensions)}")
    print(f"   - الحد الأقصى للأبعاد: {np.max(dimensions)}")


    # 7. استرجاع حالة من SQL
    print("\n6️⃣ استرجاع حالة كمومية من SQL...")
    with sqlite3.connect("periodic_table.db") as conn:
        conn.row_factory = sqlite3.Row
        cursor = conn.cursor()
        cursor.execute("""
            SELECT atomic_number, state_vector, dimension
            FROM quantum_states
            WHERE atomic_number = 1
            ORDER BY created_at DESC LIMIT 1
        """)
        row = cursor.fetchone()
        if row:
            state = np.array(json.loads(row['state_vector']))
            print(f"   ✅ استرجاع الحالة الكمومية للعنصر {row['atomic_number']}")
            print(f"   - الأبعاد: {row['dimension']}")
            print(f"   - المتجه: {state[:5]}...")


    print("\n" + "=" * 60)
    print("✅ اكتمل تشفير الحالات الكمومية")
    print("=" * 60)


if __name__ == "__main__":
    example_usage()
```

11.4 شرح الكود

Code Explanation

1. اختيار الخصائص (select_quantum_features): نختار الخصائص الرقمية التي ستمثل أبعاد الحالة الكمومية. هذه الخصائص يجب أن تكون ذات معنى فيزيائي وكيميائي.

1. Feature selection chooses numerical properties with an explicit physical or chemical interpretation.

2. التطبيع (normalize_to_quantum_state): نقسم المتجه على معياره (Norm) لضمان أن مربع معاملاته يساوي 1، وهو شرط أساسي للحالات الكمومية الصالحة.

2. Normalization makes the vector satisfy the mathematical normalization condition required of a pure state representation.

3. مصفوفة الكثافة (vector_to_density_matrix): للحالات النقية، نستخدم np.outer لتحويل المتجه إلى مصفوفة كثافة. هذا التمثيل مفيد في الحوسبة الكمومية للتعامل مع الأنظمة المختلطة والقياس.

3. A density matrix can represent a pure state as $|\psi\rangle\langle\psi|$; it should not be confused with a measured density of matter.

4. فئة QuantumStateEncoder: هي الواجهة الرئيسية لتشفير الخواص إلى حالات كمومية. توفر دوال لتشفير عنصر واحد (encode_element)، وتشفير مجموعة (encode_batch)، وتشفير جميع العناصر (encode_all_elements).

4. QuantumStateEncoder provides a reproducible software boundary between classical feature vectors and their abstract quantum-state representation.

5. حفظ واسترجاع الحالات (save_quantum_states, load_quantum_states): نستخدم np.savez_compressed لحفظ الحالات بتنسيق مضغوط، مما يوفر مساحة تخزين.

5. np.savez_compressed provides compact persistence for the encoded vectors.

6. تخزين الحالات في SQL (store_quantum_states_in_sql): نخزن الحالات الكمومية كـ JSON في قاعدة البيانات، مع إصدار (version) لتتبع التغييرات.

6. Store quantum states in SQL as JSON, together with a version field for traceability.

11.5 تمارين وتطبيقات

11.5 Exercises

1. قم بتشفير عنصر الكربون (العدد الذري 6) كحالة كمومية، واحسب تشابه جيب التمام (Cosine Similarity) بينه وبين عنصر السيليكون (العدد الذري 14).

1. Encode carbon (atomic number 6) and silicon (atomic number 14) with the same feature definition, then compare their cosine similarity.

2. قم بإنشاء مصفوفة تشابه بين جميع العناصر بناءً على حالاتها الكمومية (استخدم المسافة الإقليدية أو تشابه جيب التمام).

2. Build an element-by-element similarity matrix using Euclidean distance or cosine similarity.

3. استخدم PCA على مصفوفة الحالات الكمومية (118 عنصر × n أبعاد) ورسم أول مكونين رئيسيين.

3. Apply PCA to the encoded-state matrix and inspect the first two components.

4. أضف خصائص إضافية (مثل التوزيع الإلكتروني، حالات الأكسدة) كأبعاد إضافية في الحالة الكمومية وقارن النتائج.

4. Add electron-configuration or oxidation-state features only after defining a reproducible numerical encoding.

5. قم بتصدير الحالات الكمومية بتنسيق مناسب لاستخدامها في إطار عمل Qiskit (مثل Statevector).

5. Export the encoded vectors to a format that a chosen quantum framework can consume, while documenting that this is an abstract feature encoding.

الفصل الثاني عشر: محاكاة ذرة الهيدروجين باستخدام VQE

Chapter 12: VQE Hydrogen Simulation

12.1 مقدمة

Introduction

‏(Variational Quantum Eigensolver (VQE هي خوارزمية كمومية هجينة (تستخدم كلاً من الحاسوب الكلاسيكي والكمي) لحساب الطاقة الأساسية (Ground State Energy) للأنظمة الكمومية، مثل الذرات والجزيئات.

The Variational Quantum Eigensolver (VQE) is a hybrid quantum-classical algorithm that estimates ground-state energies of quantum systems such as atoms and molecules.

في هذا الفصل سنستخدم جزيء الهيدروجين H_2، لا ذرة الهيدروجين المفردة، لأن VQE في هذا المثال يحتاج إلى مسألة بنية إلكترونية وهاملتونيًا مشتقًا من نموذج فيزيائي واضح. ستأتي بيانات الجدول الدوري هنا بوصفها سياقًا ومصدرًا للهوية والتتبع، بينما يُشتق الهاملتوني من حساب بنية جزيئية مستقل.

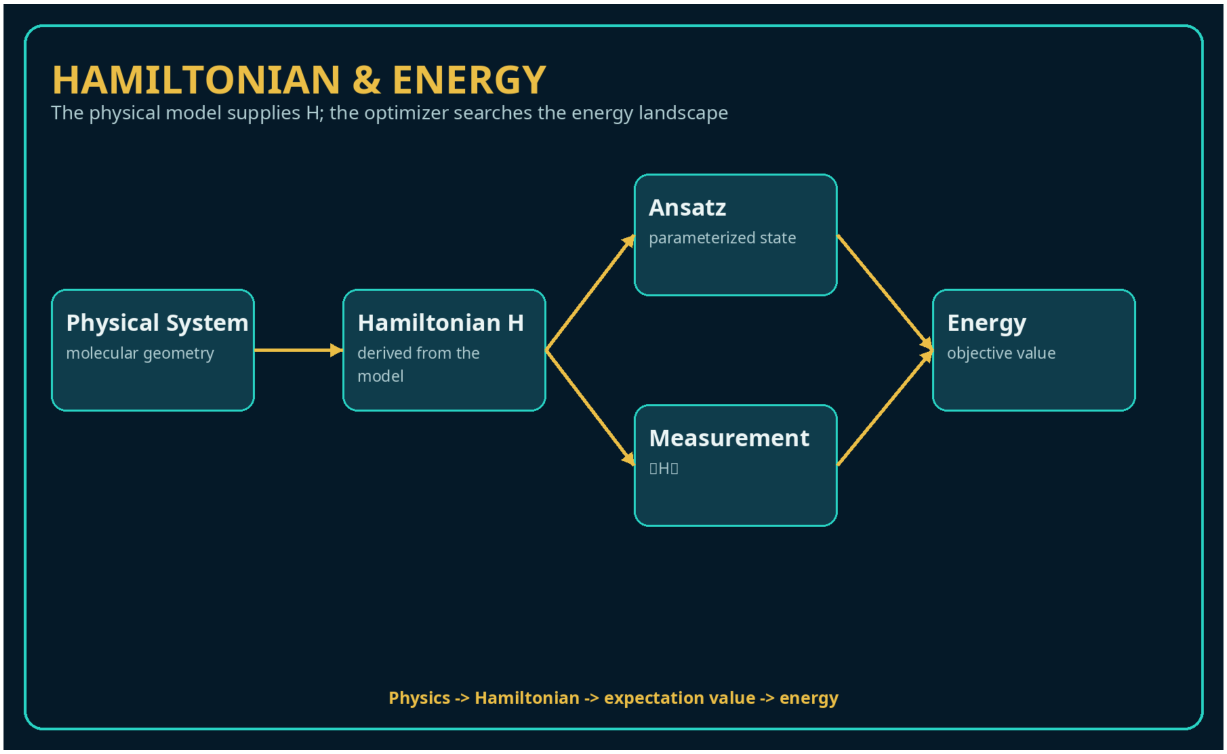
In this chapter we use molecular hydrogen, H_2, rather than an isolated hydrogen atom, because a reproducible VQE example should start from a well-defined electronic-structure problem and its derived Hamiltonian. Periodic-table data provide identity, context, and provenance; the molecular Hamiltonian is generated by the electronic-structure workflow.

12.2 الأساس النظري

12.2 Theoretical Foundation

الهاميلتوني والطاقة

Hamiltonian and Energy



توضح اللوحة العلاقة بين النظام الفيزيائي ومنظر الطاقة والهاميلتوني والقياس، مع إبقاء النموذج الفيزيائي هو مصدر الهاملتوني بدل اختراعه من خصائص جدولية.

This plate links the physical system, energy landscape, Hamiltonian, and measurement while keeping the physical model as the source of the Hamiltonian rather than inventing one from tabular features.

هاميلتوني الهيدروجين:

Hydrogen Hamiltonian:

هاملتوني النظام الإلكتروني

Electronic Hamiltonian

البرمجي H_2 في النموذج الذري البسيط يمكن كتابة مؤثر الهاملتوني على صورة طاقة حركة وطاقة وضع. أما في مثال التكاملات الإلكترونية، PySCF هنا، فلا نضع هذه الصيغة يدويًا ولا نشتقها من نصف القطر أو طاقة التأين؛ بل يحسب إلى مؤثر كمومي مناسب Qiskit Nature ثم يحولها.

In the H_2 example used by the code, the Hamiltonian is not fabricated from atomic radius or ionization energy. PySCF computes the electronic integrals from the molecular geometry and basis set,

and Qiskit Nature converts the resulting second-quantized problem into a qubit operator.

حيث يوضح التعبير السابق الفكرة الفيزيائية العامة لطاقة الإلكترون في مجال النواة. لكن الكود التنفيذي في هذا الفصل يستخدم الهاملتوني الجزيئي الناتج عن حساب H_2، وهو المسار الصحيح لإجراء تجربة VQE قابلة لإعادة البناء.

The preceding expression illustrates the physical idea of an electron moving in a nuclear potential. The executable experiment in this chapter instead uses the molecular H_2 Hamiltonian produced by the electronic-structure calculation, which makes the VQE workflow reproducible.

الخطوات الأساسية لـ VQE:

Core VQE Steps:

1. بناء الهاملتوني (Hamiltonian) للنظام.

1. Build or obtain the physical Hamiltonian for the molecular problem.

2. تصميم دائرة أنساتز (Ansatz Circuit) – دائرة كمومية قابلة للتعديل بمعاملات.

2. Prepare a parameterized ansatz circuit.

3. حساب القيمة المتوقعة للهاملتوني على الحالة التي تنتجها الدائرة.

3. Estimate the Hamiltonian expectation value for the circuit state.

4. استخدام محسن كلاسيكي (Classical Optimizer) لتعديل معاملات الدائرة لتقليل الطاقة.

4. Use a classical optimizer to adjust circuit parameters and reduce the estimated energy.

5. تكرار الخطوات حتى الوصول إلى الطاقة الأساسية.

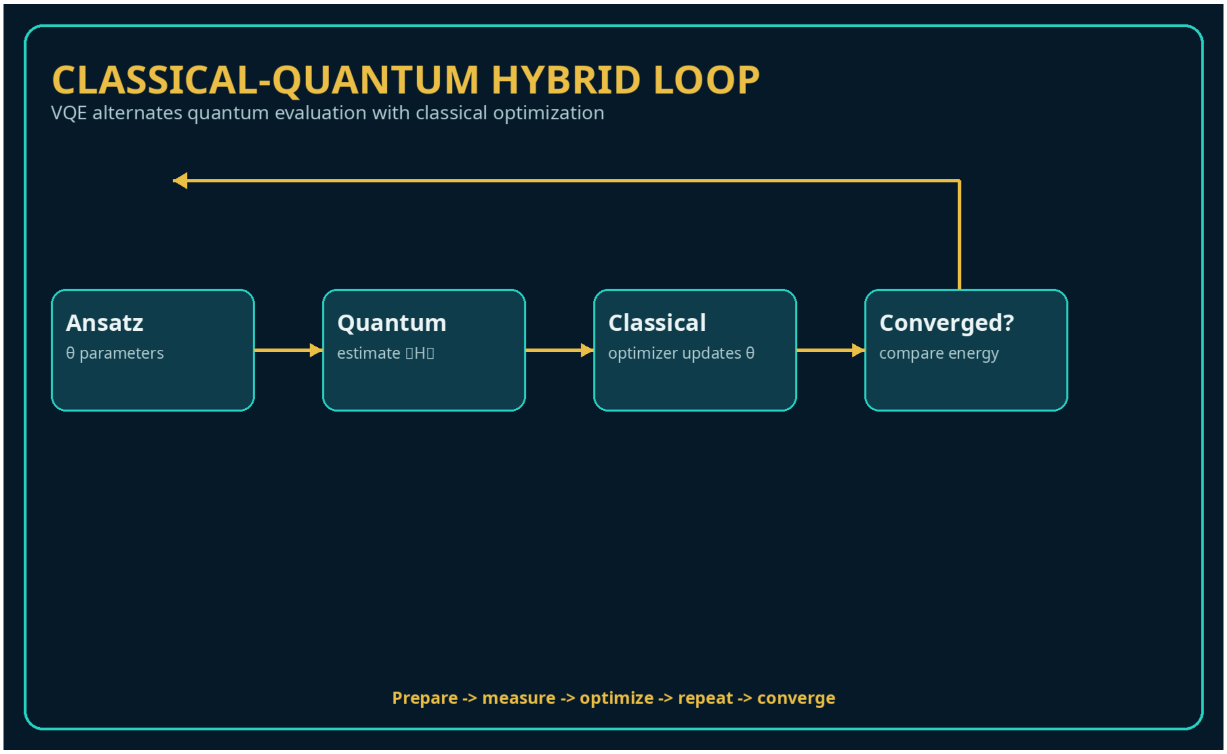
5. Repeat the loop until the optimization converges, then compare the result with an independent reference.

12.3 كود محاكاة VQE للهيدروجين

VQE Hydrogen Simulation Code

الحلقة الهجينة الكلاسيكية–الكمومية

Classical–Quantum Hybrid Loop



توضح اللوحة حلقة VQE: دائرة كمومية، تقدير للطاقة، محسّن كلاسيكي، ثم إعادة التقييم حتى التقارب، مع حفظ النتيجة في مسار قابل للتكرار.

This plate shows the VQE loop: a quantum circuit, energy estimation, a classical optimizer, and repeated evaluation until convergence, with the result retained in a reproducible workflow.

VQE Hydrogen Simulation Code

PYTHON

```python
# ============================================================
# vqe_h2_simulation.py
# محاكاة جزيء الهيدروجين H₂ باستخدام VQE وQiskit Nature
# Ground-state VQE for molecular hydrogen (H₂)
# ============================================================

from qiskit_nature.units import DistanceUnit
from qiskit_nature.second_q.drivers import PySCFDriver
from qiskit_nature.second_q.mappers import ParityMapper
from qiskit_nature.second_q.circuit.library import HartreeFock, UCCSD
from qiskit_nature.second_q.algorithms import GroundStateEigensolver
from qiskit_algorithms import VQE
```

```
from qiskit_algorithms.optimizers import L_BFGS_B
from qiskit.primitives import StatevectorEstimator


def build_h2_problem():
    """Build an H₂ electronic-structure problem with PySCF."""
    driver = PySCFDriver(
        atom="H 0.0 0.0 0.0; H 0.0 0.0 0.735",
        unit=DistanceUnit.ANGSTROM,
        basis="sto3g",
    )
    return driver.run()


def run_vqe():
    """Run VQE and return the Qiskit Nature electronic-structure result."""
    problem = build_h2_problem()

    mapper = ParityMapper(num_particles=problem.num_particles)
    initial_state = HartreeFock(
        problem.num_spatial_orbitals,
        problem.num_particles,
        mapper,
    )
    ansatz = UCCSD(
        problem.num_spatial_orbitals,
        problem.num_particles,
        mapper,
        initial_state=initial_state,
    )

    estimator = StatevectorEstimator()
    optimizer = L_BFGS_B()
    vqe = VQE(estimator, ansatz, optimizer)
    vqe.initial_point = [0.0] * ansatz.num_parameters

    solver = GroundStateEigensolver(mapper, vqe)
    result = solver.solve(problem)
    result.formatting_precision = 6
    return result


if __name__ == "__main__":
    result = run_vqe()
    print(result)
```

12.4 شرح الكود

Code Explanation

1. بناء المسألة الكيميائية (build_h2_problem): يستخدم PySCF هندسة H_2 عند مسافة 0.735 Å وقاعدة STO-3G لبناء مسألة البنية الإلكترونية من التكاملات المحسوبة، بدل اختراع هاملتوني من خواص الجدول الدوري.

1. Building the chemistry problem: PySCF uses the H_2 geometry at 0.735 Å with the STO-3G basis to construct the electronic-structure problem from computed integrals instead of inventing a Hamiltonian from periodic-table properties.

2. الحالة الابتدائية والـAnsatz: تُستخدم HartreeFock كحالة ابتدائية، ثم UCCSD كدالة موجية متغيرة مناسبة لمسائل البنية الإلكترونية.

2. Initial state and ansatz: Hartree–Fock supplies the initial state, while UCCSD provides a chemistry-oriented variational wavefunction for the electronic-structure problem.

3. مقوّم الطاقة: StatevectorEstimator هو تطبيق مرجعي محلي لـEstimator V2 ويحسب قيم التوقع للحالة والملاحظات دون افتراض وجود جهاز كمومي فعلي.

3. Energy estimator: StatevectorEstimator is Qiskit's local V2 estimator implementation and evaluates expectation values without requiring a real quantum device.

4. المحسن الكلاسيكي: L_BFGS_B يغيّر معاملات UCCSD بهدف تقليل الطاقة المتوقعة، وهو الجزء الكلاسيكي من الحلقة الهجينة.

4. Classical optimizer: L_BFGS_B adjusts the UCCSD parameters to minimize the estimated energy; this is the classical component of the hybrid loop.

5. الحل والتوثيق: GroundStateEigensolver يربط QubitMapper مع VQE ويعيد نتيجة بنية إلكترونية قابلة للحفظ والتدقيق.

5. Solution and provenance: GroundStateEigensolver connects the qubit mapper to VQE and returns an electronic-structure result that can be stored and audited.

6. لا ينبغي مقارنة الناتج مع قيمة ذرة الهيدروجين ‎-0.5 Hartree؛ المثال يحسب جزيء H_2، ولذلك تُقارن النتيجة مع مرجع مسألة H_2 نفسها، ويمكن استخدام محلل مصفوفة كلاسيكي مستقل للتحقق.

6. Do not compare this result with the isolated hydrogen-atom value of -0.5 Hartree; the example computes molecular H_2. Compare against an independent H_2 reference calculation instead.

12.5 تمارين وتطبيقات

Exercises

1. غير عمق دائرة الأناتز (depth) من 2 إلى 4 وشاهد تأثير ذلك على دقة الطاقة.

1. Change the ansatz circuit depth from 2 to 4 and observe its effect on the estimated energy.

2. حاول تشغيل VQE لعنصر الهيليوم (العدد الذري 2) باستخدام 2 كيوبت. (ستحتاج إلى تعديل الهاملتوني)

2. Try running VQE for helium (atomic number 2) with two qubits; this requires a physically defined Hamiltonian for the chosen problem.

3. استخدم محسنًا مختلفًا (مثل SPSA من qiskit.algorithms.optimizers) وقارن النتائج.

3. Try another optimizer and compare its convergence behavior. If using a current Qiskit release, use the optimizer API documented for that release.

4. أضف قياسات إضافية (مثل الحالة النهائية، توزيع الاحتمالات) إلى النتائج المخزنة.

4. Store additional observables, final-state information, or probability distributions with the run record.

5. قم بإنشاء رسم بياني يوضح تطور الطاقة خلال تكرارات المحسن.

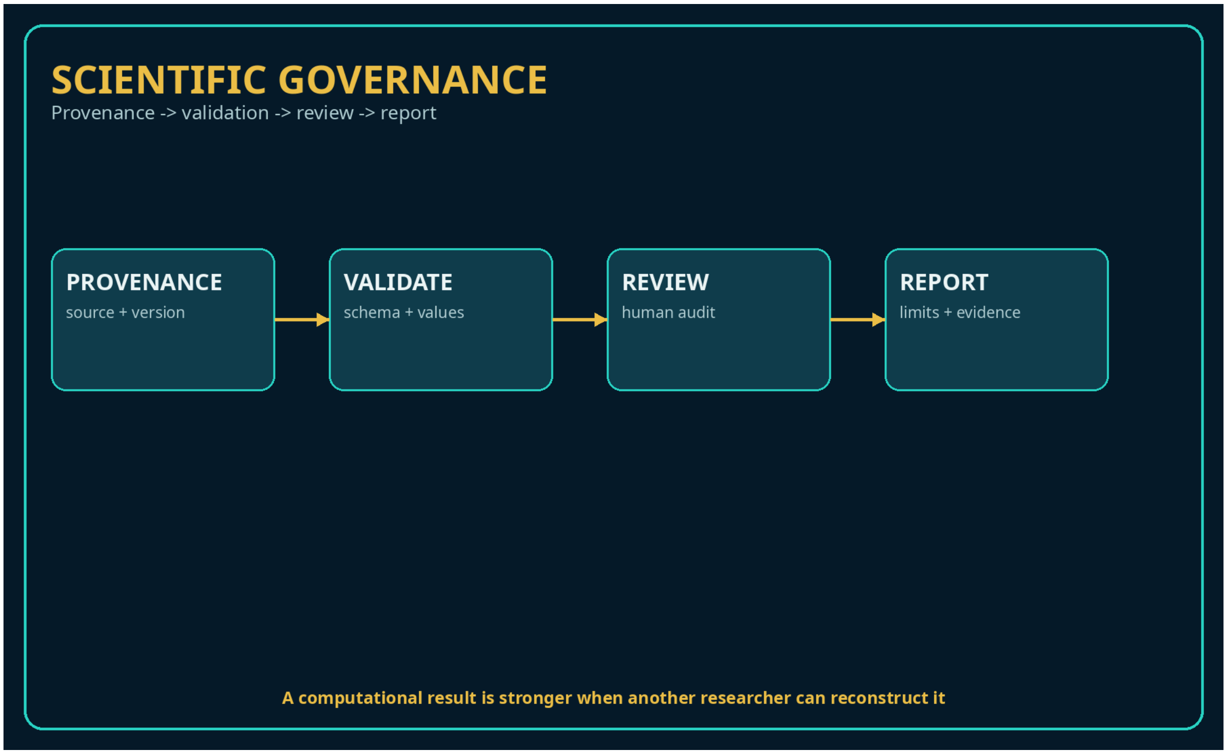
5. Plot the energy across optimizer iterations.

الفصل الثالث عشر: تخزين النتائج الكمومية في SQL – Storing Quantum Results in SQL

Chapter 13: Storing Quantum Results in SQL

الحوكمة العلمية

Scientific Governance



تربط اللوحة النتيجة العلمية بالمصدر والتحقق والمراجعة والتقرير، بحيث تصبح النتيجة الحسابية قابلة لإعادة البناء والمساءلة.

This plate connects a scientific result to provenance, validation, review, and reporting so the computational result remains reconstructable and accountable.

13.1 لماذا نخزن النتائج؟

Why Store Quantum Results?

نتيجة تجربة كمومية بلا سياق لا تكفي. نحتاج إلى معرفة النظام، والهاملتوني، والدائرة، والمعلمات، والمحسن، وإصدار البرمجيات، ووقت التشغيل.

A quantum result without context is difficult to reproduce. Store the system, Hamiltonian, circuit, parameters, optimizer, software version, and execution timestamp alongside the numerical result.

13.2 مخطط نتائج VQE

VQE Results Schema

SQL

```
CREATE TABLE quantum_runs (
    run_id INTEGER PRIMARY KEY AUTOINCREMENT,
    system_name TEXT NOT NULL,
    hamiltonian TEXT NOT NULL,
    backend TEXT NOT NULL,
    optimizer TEXT,
    energy REAL,
    parameters_json TEXT,
    software_version TEXT,
    created_at TEXT DEFAULT CURRENT_TIMESTAMP
);
```

يفصل الجدول هوية التجربة عن نتيجتها العددية وبيانات مصدرها. ويمكن لـ JSONحفظ متجهات المعاملات بينما تبقى الحقول العلائقية الأساسية قابلة للاستعلام.

The table separates an experiment identity from its numerical result and provenance. JSON can hold parameter vectors while the relational columns retain the fields most often queried.

13.3 إدخال نتيجة مع معاملة

Transactional Storage

SQL

```
BEGIN TRANSACTION;
INSERT INTO quantum_runs
(system_name, hamiltonian, backend, optimizer, energy, parameters_json, software_version)
VALUES (?,?,?,?,?,?,?);
COMMIT;
```

استخدم SQL بمعاملات محددة مسبقًا ومعاملات Transaction. تمنع المعاملات أخطاء تركيب النصوص، وتحافظ المعاملة على ذرّية الكتابة عند تخزين عدة سجلات معًا.

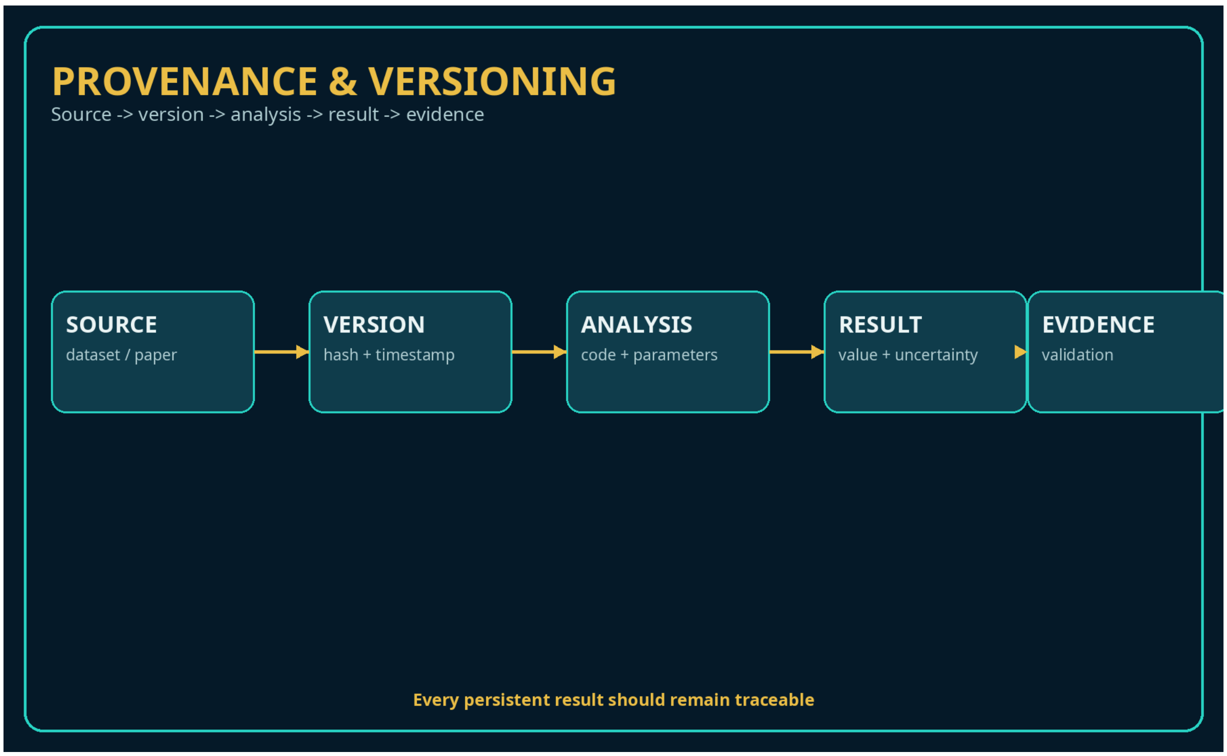
Use parameterized SQL and transactions. Parameterization avoids string-construction errors and transactions preserve atomicity when several records must be written together.

13.4 الربط بين التجارب

Comparing Runs

المصدر والإصدار والتتبع

Provenance and Versioning



توضح اللوحة كيف ترتبط النسخة بالمصدر والتحليل والنتيجة والدليل، وهو ما ينسجم مباشرة مع تخزين نتائج VQE وإصدارات البرمجيات والطوابع الزمنية.

This plate shows how a version connects to its source, analysis, result, and evidence, directly supporting the provenance requirements of persistent VQE records.

Comparing Runs

SQL

```
SELECT system_name, backend, MIN(energy) AS best_energy
FROM quantum_runs
```

```
GROUP BY system_name, backend
ORDER BY best_energy;
```

مقارنة التجارب وكشف أنماط إعادة الإنتاج دون أن يحل محل التفسير العلمي للتجربة SQL بعد حفظ النتائج، يستطيع.

Once results are persistent, SQL can compare runs and expose reproducibility patterns without replacing the scientific interpretation of the experiment.

13.5 تمارين وتطبيقات

Exercises

1. أضف حقلًا circuit_hash لتتبع نسخة الدائرة.

1. Add a circuit_hash field to identify circuit versions.

2. خزّن seed الخاص بالمحسن.

2. Store the optimizer seed.

3. أنشئ استعلامًا يعرض أفضل نتيجة لكل إصدار برمجي.

3. Query the best result for each software version.

الفصل الرابع عشر: التكامل التشغيلي — من البيانات إلى القرار

Chapter 14: Operational Integration — From Data to Decision

هذا الفصل يجمع المسارات التي ظهرت متفرقة في المشروع ويحوّلها إلى بنية واحدة: بيانات الجدول الدوري تدخل من قاعدة البيانات، ثم تمر عبر طبقات التحليل والتعلّم الآلي والتوليد، ويمكن أن تصل إلى تمثيل كمومي عندما توجد مسألة مناسبة. الهدف ليس إضافة تقنيات لمجرد كثرتها، بل توضيح متى تستخدم كل طبقة، وما الذي تستطيع إثباته، وما الذي لا تستطيع إثباته.

This chapter joins the project's previously separate paths into one architecture: periodic-table data enter through the database, pass through numerical analysis and machine learning, may feed generative workflows, and can reach a quantum representation when a suitable problem exists. The goal is not to accumulate

technologies, but to state when each layer is justified, what it can establish, and what it cannot establish.

14.1 لماذا نحتاج طبقة تكامل؟

14.1 Why an Integration Layer Is Necessary

عندما تعمل كل تقنية وحدها، يسهل أن تبدو النتيجة أقوى مما هي عليه. نموذج تعلم آلي قد يتنبأ، لكنه لا يثبت السببية. نموذج توليدي قد يقترح مركبًا، لكنه لا يثبت استقراره أو قابليته للتصنيع. محاكاة كمومية قد تقدّر كمية لمشكلة محددة، لكنها لا تحوّل بيانات الجدول الدوري إلى وصف كامل لمادة حقيقية. لذلك نحتاج طبقة تكامل تحفظ العلاقة بين المدخلات، والتحويلات، والنتائج، وحدود الادعاء.

When each technology is considered in isolation, a result can appear stronger than it really is. A machine-learning model may predict without establishing causality. A generative model may propose a compound without proving stability or synthesizability. A quantum simulation may estimate a quantity for a defined problem without turning periodic-table data into a complete description of a real material. The integration layer therefore preserves the relationship between inputs, transformations, outputs, and claim limits.

14.2 الذكاء الاصطناعي التوليدي ونظام التوصية

14.2 Generative AI and a Recommendation Layer

يمكن استخدام GANs أو VAEs لاستكشاف فضاء مرشحين لمواد أو تراكيب، ويمكن استخدام نماذج لغوية أو نماذج تحويلية لتوليد تمثيلات نصية أو اقتراحات تركيبية. لكن جدول العناصر وحده لا يكفي لتدريب نموذج لغوي كبير ذا معنى علمي؛ عدد السجلات 118 فقط، ولذلك يجب النظر إلى الجدول الدوري هنا كطبقة مرجعية وميزة أساسية، ثم ربطه بمجموعات بيانات موثقة أكبر إذا كان الهدف بناء نموذج توليدي فعلي.

GANs or VAEs can be used to explore candidate spaces for materials or compositions, and language or transformer models can generate textual representations or candidate suggestions. The 118-element table alone is not enough to train a scientifically meaningful large language model. It should instead serve as a reference and feature layer, connected to larger, documented datasets when a genuine generative model is required.

نظام التوصية أبسط وأكثر قابلية للتدقيق في هذا المثال: نعرّف مجموعة خواص مستهدفة، ثم نحسب قرب العناصر أو التركيبات المرشحة من هذا الهدف. بهذه الطريقة يمكن فصل «اقتراح المرشح» عن «الحكم العلمي على المرشح».

A recommendation layer is simpler and more auditable in this project: define a target property vector, then calculate the similarity of candidate elements or compositions to that target. This separates candidate recommendation from scientific judgment about the candidate.

PYTHON

```
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import NearestNeighbors

# X contains documented element features: rows = elements, columns = features.
# target contains the desired property profile in the same feature order.
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
target_scaled = scaler.transform(np.asarray(target).reshape(1, -1))

model = NearestNeighbors(n_neighbors=5, metric='euclidean')
model.fit(X_scaled)
distance, index = model.kneighbors(target_scaled)

recommendations = [(int(i), float(d)) for i, d in zip(index[0], distance[0])]
print(recommendations)
```

هذا الكود لا «يكتشف مادة جديدة»؛ إنه ينفذ طبقة توصية قابلة لإعادة الإنتاج على متجهات ميزات موثقة. إذا كانت الميزات غير ممثلة تمثيلًا صحيحًا أو كانت البيانات متحيزة، فستكون التوصيات متحيزة كذلك.

This code does not “discover a new material.” It implements a reproducible recommendation layer over documented feature vectors. If the features are poorly represented or the dataset is biased, the recommendations will be biased as well.

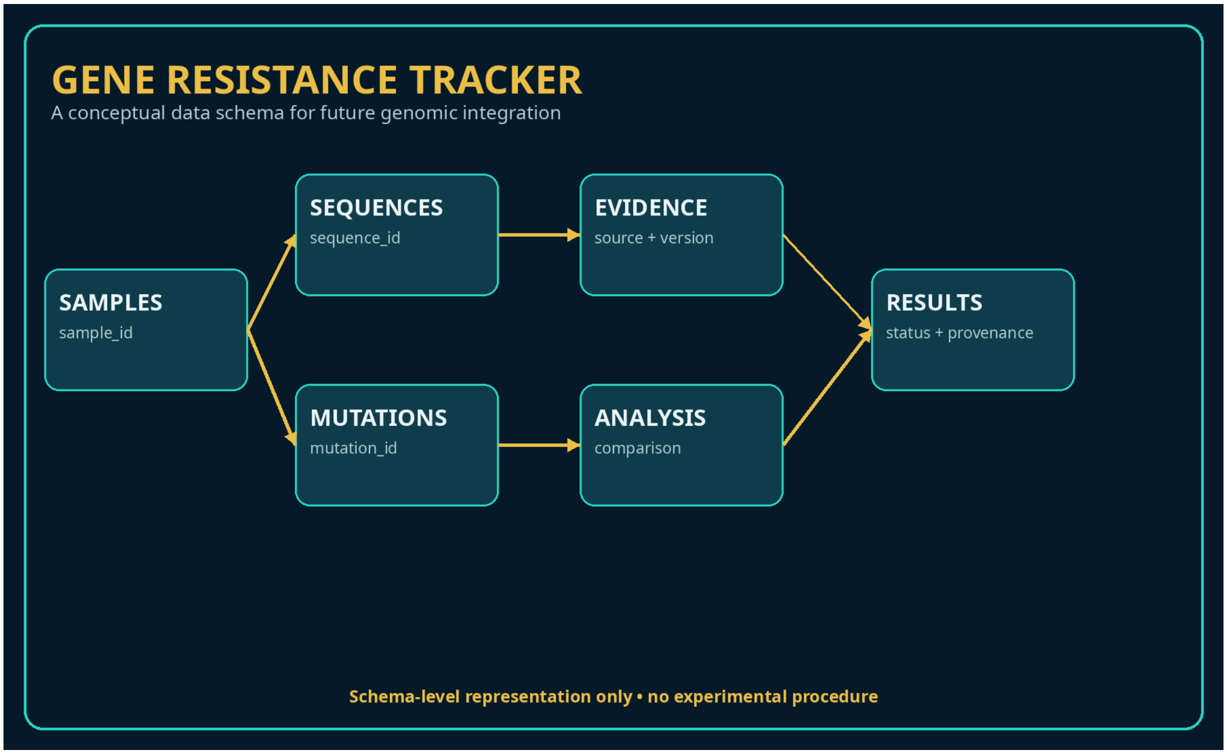
14.3 الجسر إلى البيانات الجينومية — Gene Resistance Tracker

14.3 The Bridge to Genomic Data — Gene Resistance Tracker متتبع مقاومة الجينات

Gene Resistance Tracker

Gene Resistance Tracker

Gene Resistance Tracker



يمكن ربط بيانات العناصر ببيانات البروتينات والإنزيمات، ثم بكيانات العينات والتسلسلات والطفرات والأدلة. لكن العلاقة يجب أن تكون علاقة بيانات موثقة، لا قفزة سببية: وجود عنصر داخل بنية بروتينية أو مسار أيضي لا يثبت أن هذا العنصر سبب مرضًا أو مقاومة دوائية.

Element data can be linked to proteins and enzymes and then to samples, sequences, variants, and evidence. The relationship must remain a documented data relationship, not a causal leap: the presence of an element in a protein structure or metabolic pathway does not by itself prove that the element causes a disease or drug resistance.

تصبح هوية المصدر والإصدار جزءًا من النموذج نفسه. كل ملاحظة مقاومة في Gene Resistance Tracker يجب أن تشير إلى الجين أو المتغير، ومصدر الدليل، وإصدار البيانات، ومستوى الثقة أو الدليل. هذا يمنع أن تتحول نتيجة تجميعية مجهولة المصدر إلى «حقيقة» في النظام.

In the Gene Resistance Tracker, provenance and version become part of the data model itself. Each resistance observation should reference the gene or variant, evidence source, dataset version, and

an evidence or confidence level. This prevents an unattributed aggregate result from becoming a false “fact” inside the system.

SQL

```
CREATE TABLE genomic_links (
    link_id INTEGER PRIMARY KEY,
    atomic_number INTEGER NOT NULL,
    protein_id TEXT,
    enzyme_id TEXT,
    evidence_level TEXT NOT NULL,
    source_version TEXT NOT NULL,
    FOREIGN KEY (atomic_number) REFERENCES elements(atomic_number)
);

CREATE TABLE resistance_observations (
    observation_id INTEGER PRIMARY KEY,
    gene_id TEXT NOT NULL,
    variant_id TEXT,
    phenotype TEXT NOT NULL,
    evidence_level TEXT NOT NULL,
    source_version TEXT NOT NULL
);
```

هذه الجداول أمثلة تعليمية ببيانات معرفات، وليست بيانات مرضى ولا بروتوكولًا تجريبيًا. الغرض هو إظهار كيف يمكن أن يصبح التكامل الجينومي امتدادًا منظمًا للطبقات التي بُنيت في المشروع.

These tables are educational examples using identifiers; they are not patient data and are not an experimental protocol. Their purpose is to show how genomic integration can extend the project’s existing layers in a structured way.

14.4 المحاكاة الكمومية متعددة الكيوبت: 4–6 كيوبت وما بعدها

14.4 Multi-Qubit Quantum Simulation: 4–6 Qubits and Beyond

يحتاج الانتقال من H_2 إلى أنظمة مثل H_2O أو CO_2 إلى أكثر من زيادة عدد الكيوبتات. يجب تحديد الهندسة الجزيئية، وقاعدة الأساس، ونوع التقريب الإلكتروني، ثم بناء الهاملتوني، وبعد ذلك اختيار ترميز كيوبتي ودائرة مناسبة. دوائر 4–6 كيوبت مفيدة هنا كمنطقة تعليمية صغيرة لفهم التوسع، لكنها لا تعني أن نظامًا كاملًا متعدد الإلكترونات قد تم تمثيله بدقة كيميائية كاملة.

Moving from H_2 to systems such as H_2O or CO_2 requires more than increasing the qubit count. Molecular geometry, basis set, electronic-structure approximation, and Hamiltonian must be defined first, followed by a qubit mapping and a suitable circuit. Four-to-six-qubit circuits are useful as a small educational scaling

regime, but they do not imply a fully accurate representation of a many-electron molecule.

يبقى المسار الهجين هو الفكرة الأساسية: الحاسوب الكلاسيكي يختار المعاملات، والدائرة الكمومية تقدّر القيمة المتوقعة، ثم تعود النتيجة إلى المحسّن. زيادة عدد الكيوبتات تزيد مساحة الحالة بسرعة، كما أن الضوضاء وعمق الدائرة وعدد القياسات تصبح قيودًا عملية مستقلة عن حجم قاعدة البيانات. عند استخدام VQE،

With VQE, the hybrid loop remains the central idea: the classical optimizer chooses parameters, the quantum circuit estimates an expectation value, and the result returns to the optimizer. Increasing the qubit count expands the state space rapidly, while noise, circuit depth, and measurement cost become practical constraints independent of database size.

أما HHL فيجب تقديمه بدقة: هو خوارزمية لحل أنظمة خطية كمومية تحت افتراضات محددة، وليس زرًا مباشرًا لحل معادلة شرودنجر. لاستخدامه في مسألة فيزيائية يجب أولًا تحويل المسألة إلى نظام خطي مناسب، والتعامل مع شروط مثل تحضير الحالة، وبنية المصفوفة، وعدد الشرط، واستخراج النتيجة من الحالة الكمومية.

HHL should be described precisely: it is a quantum algorithm for solving linear systems under specific assumptions, not a direct button for solving the Schrödinger equation. A physical application first requires a suitable linear-system formulation and attention to state preparation, matrix structure, condition number, and extraction of useful information from the quantum state.

14.5 التوسع الحسابي: Dask وSpark والتعلم العميق

14.5 Computational Scaling: Dask, Spark, and Deep Learning

لا تبدأ بالتوزيع قبل أن تثبت أن المشكلة تحتاجه. إذا كان جدول العناصر 118 صفًا، فإن NumPy وSQLite وscikit-learn تكفي غالبًا. يصبح Dask أو Spark منطقيًا عندما تتجاوز البيانات الذاكرة، أو عندما تتكرر عمليات مكلفة على عدد كبير من السجلات. وبالمثل، لا تحتاج إلى TensorFlow أو PyTorch لمجرد وجود كلمة «ذكاء اصطناعي»؛ استخدم التعلم العميق عندما يكون حجم البيانات وبنية المشكلة يبرران ذلك.

Do not distribute a workload before demonstrating that it needs distribution. For 118 element records, SQLite, NumPy, and scikit-learn are generally sufficient. Dask or Spark become justified

when data no longer fit comfortably in memory or repeated computation becomes expensive. Likewise, PyTorch or TensorFlow should not be introduced merely because the phrase “AI” appears; deep learning is justified when the dataset size and problem structure support it.

هذه النقطة مهمة لأن الأداء ليس سباقًا بين الأدوات. الأداة الأفضل هي التي تحقق الهدف بأقل تعقيد يمكن تدقيقه. عندما تكبر البيانات الجينومية فعلًا، يمكن أن تنتقل المصفوفات من الذاكرة المحلية إلى معالجة دفعات، ثم إلى بيئة موزعة، مع الاحتفاظ بنفس تعريف الميزات ونفس سجل الإصدارات.

This matters because performance is not a contest between tools. The best tool is the one that achieves the goal with the least complexity that can still be audited. When genomic data genuinely become large, matrices can move from local memory to batch processing and then to distributed execution while preserving the same feature definitions and version records.

14.6 الحوكمة قبل التوليد والتنفيذ

14.6 Governance Before Generation and Execution

عندما تنتج طبقات الذكاء الاصطناعي أو الحوسبة الكمومية عددًا كبيرًا من المرشحين، تصبح المراجعة جزءًا من خط الأنابيب وليست خطوة شكلية في نهايته. يمكن أن تتضمن البنية بوابة موافقة مسبقة، وسجلًا للمدخلات، وإصدارات البرمجيات، وسبب قبول أو رفض النتيجة، وهوية المراجع، وربط النتيجة بالمصدر الذي بنيت عليه.

When AI or quantum layers generate many candidates, review becomes part of the pipeline rather than a formal step at the end. The architecture can include a pre-approval gate, input records, software versions, reasons for accepting or rejecting a result, reviewer identity, and a link from each result to the evidence on which it was based.

القيود الأخلاقية هنا ليست زينة فلسفية. هي قيود تشغيلية تمنع الانتقال غير المبرر من نتيجة حسابية إلى قرار ذي أثر مرتفع. وكلما ارتفع أثر القرار، ارتفعت الحاجة إلى التحقق المستقل، وقابلية التتبع، والمراجعة البشرية، والقدرة على إعادة بناء النتيجة.

The ethical constraints here are not decorative philosophy. They are operational controls that prevent an unjustified transition from a computational result to a high-impact decision. As the impact of a decision increases, so does the need for independent validation, traceability, human review, and the ability to reconstruct the result.

14.7 قائمة تحقق علمية قبل اعتبار المشروع مكتملًا

14.7 Scientific Completion Checklist

• هل لكل نتيجة مصدر بيانات وإصدار معروفان؟

• Does every result have a known data source and version?

• هل يمكن إعادة تشغيل الاستعلامات والأكواد على نفس المدخلات؟

• Can the queries and code be rerun on the same inputs?

• هل فُصل التنبؤ عن السببية، والاقتراح عن الإثبات؟

• Are prediction and causality, and proposal and evidence, kept distinct?

• هل كل تمثيل كمومي موصوف بوضوح: تمثيل ميزات مجرد أم نموذج فيزيائي؟

• Is every quantum representation clearly identified as an abstract feature encoding or a physical model?

• هل جرى اختبار النتائج الكمومية بمراجع كلاسيكية مستقلة عندما يكون ذلك ممكنًا؟.

• Are quantum results checked against independent classical references whenever possible.

• هل تخضع النتائج الحساسة للمراجعة والتتبع قبل استخدامها في قرار؟

• Are sensitive results reviewed and traceable before they influence a decision?

عند مشاركة المشروع مع قارئ آخر، يجب أن يعرف أي إصدار من Python والمكتبات استُخدم. ملفا requirements.txt وenvironment.yml في القسم التشغيلي النهائي يمنعان اختلافات صامتة في المكتبات من تغيير النتائج.

When the project is shared with another reader, the Python and library versions must be explicit. The requirements.txt and environment.yml files in the final operational section reduce silent dependency changes that could alter results.

5.7 تثبيت البيئة وقابلية إعادة التشغيل

5.7 Environment Pinning and Reproducible Execution

لأن مجموعة العناصر 118 سجلًا فقط، يجب تفسير هذه الأرقام بحذر. التحسين الإحصائي لا يحول مجموعة صغيرة إلى بيانات كبيرة، ولا يبرر تعميم الأداء خارج نطاق البيانات المستخدمة.

Because the periodic-table dataset contains only 118 records, these metrics must be interpreted cautiously. Statistical tuning does not turn a small dataset into a large one, and it does not justify generalizing performance beyond the data used.

PYTHON

```python
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import GridSearchCV, KFold
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler

pipeline = Pipeline([
    ("scale", StandardScaler()),
    ("model", RandomForestRegressor(random_state=42))
])

param_grid = {
    "model__n_estimators": [100, 300],
    "model__max_depth": [None, 10],
    "model__min_samples_leaf": [1, 2],
}

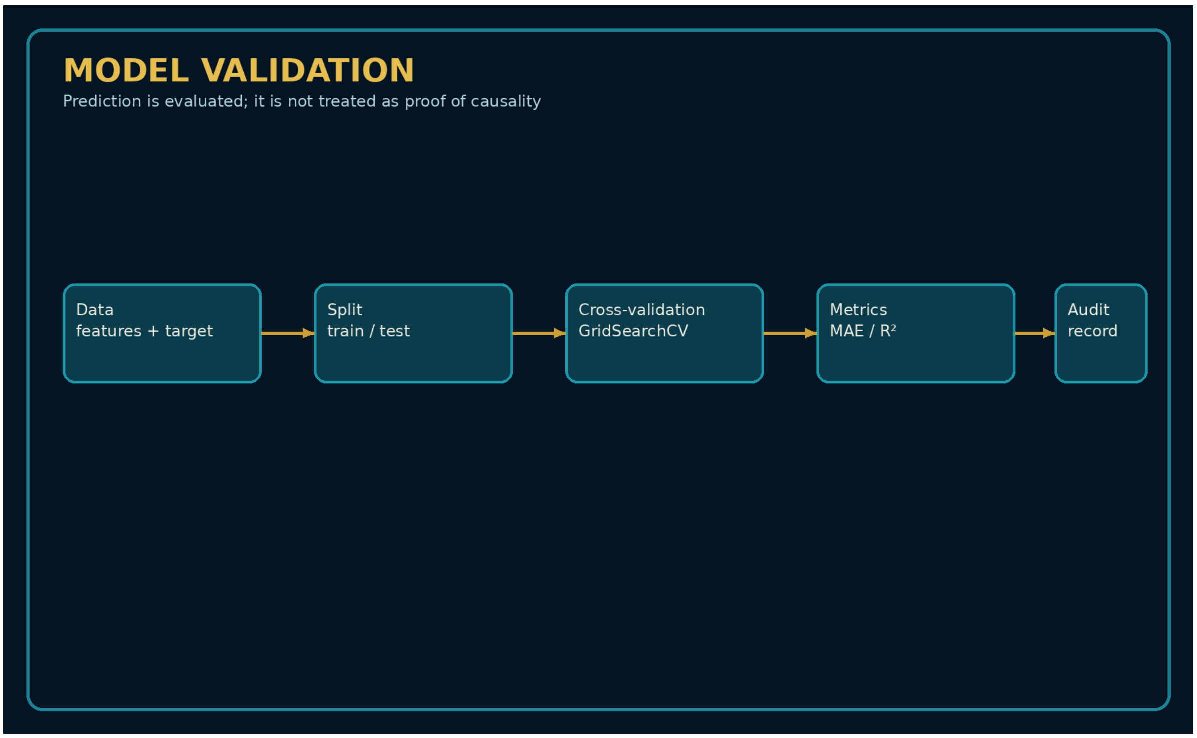
cv = KFold(n_splits=5, shuffle=True, random_state=42)
search = GridSearchCV(
    pipeline,
    param_grid=param_grid,
    scoring="neg_mean_absolute_error",
    cv=cv,
    n_jobs=-1,
)
search.fit(X_train, y_train)
print(search.best_params_)
print("CV MAE:", -search.best_score_)
```

قيمة n_estimators=300 ليست دليلًا على أن النموذج هو الأفضل. لذلك يضيف المشروع مقارنة منظمة بين إعدادات محددة باستخدام Cross-Validation، ثم يحتفظ بالمعاملات والنتيجة في سجل قابل للتدقيق.

The value n_estimators=300 is not evidence that the model is optimal. The project therefore adds a structured comparison of defined hyperparameters using cross-validation and records the selected parameters and score for auditability.

لوحة: التحقق من النموذج واختيار المعاملات

Plate: Model validation and hyperparameter selection



9.5 تحسين النموذج والتحقق المتقاطع

9.5 Model Selection and Cross-Validation

يمكن استخدام H_2O أو CO_2 كمرحلة تعليمية لاحقة، لكنهما لا يضافان بوصفهما نتائج منجزة في هذا المشروع ما لم تُحدد الهندسة، وقاعدة الأساس، ونطاق الإلكترونات، وترميز الكيوبت، وتكلفة المحاكاة، ثم تُختبر النتائج. لذلك يظل H_2 المثال التنفيذي الأساسي، بينما H_2O وCO_2 مساران توسعيان مضبوطا المتطلبات.

H_2O or CO_2 can serve as later educational extensions, but they should not be presented as completed results unless geometry, basis set, electron space, qubit mapping, simulation cost, and

validation are explicitly defined and tested. H_2 therefore remains the primary executable example, while H_2O and CO_2 are controlled extension paths.

التوسع إلى H_2O وCO_2

Extending the workflow to H_2O and CO_2

المقارنة يجب أن تسجل الفرق العددي، لا أن تختار النتيجة الأقرب إلى توقع مسبق. وإذا تجاوز الفرق الحد المقبول للمسألة المحددة، يجب أن تعود إلى إعدادات الدائرة، والمحسن، والتقريب الإلكتروني، وعدد القياسات قبل استخلاص أي استنتاج.

The comparison should record the numerical difference rather than select the result that matches a prior expectation. If the difference exceeds the tolerance defined for the problem, inspect the circuit, optimizer, electronic-structure approximation, and measurement budget before drawing a conclusion.

PYTHON

```python
from pyscf import gto, scf, fci

def classical_fci_reference():
    mol = gto.M(
        atom="H 0 0 0; H 0 0 0.735",
        unit="Angstrom",
        basis="sto-3g",
        charge=0,
        spin=0,
    )
    mf = scf.RHF(mol).run()
    solver = fci.FCI(mol, mf.mo_coeff)
    electronic_energy, _ = solver.kernel()
    return electronic_energy + mol.energy_nuc()

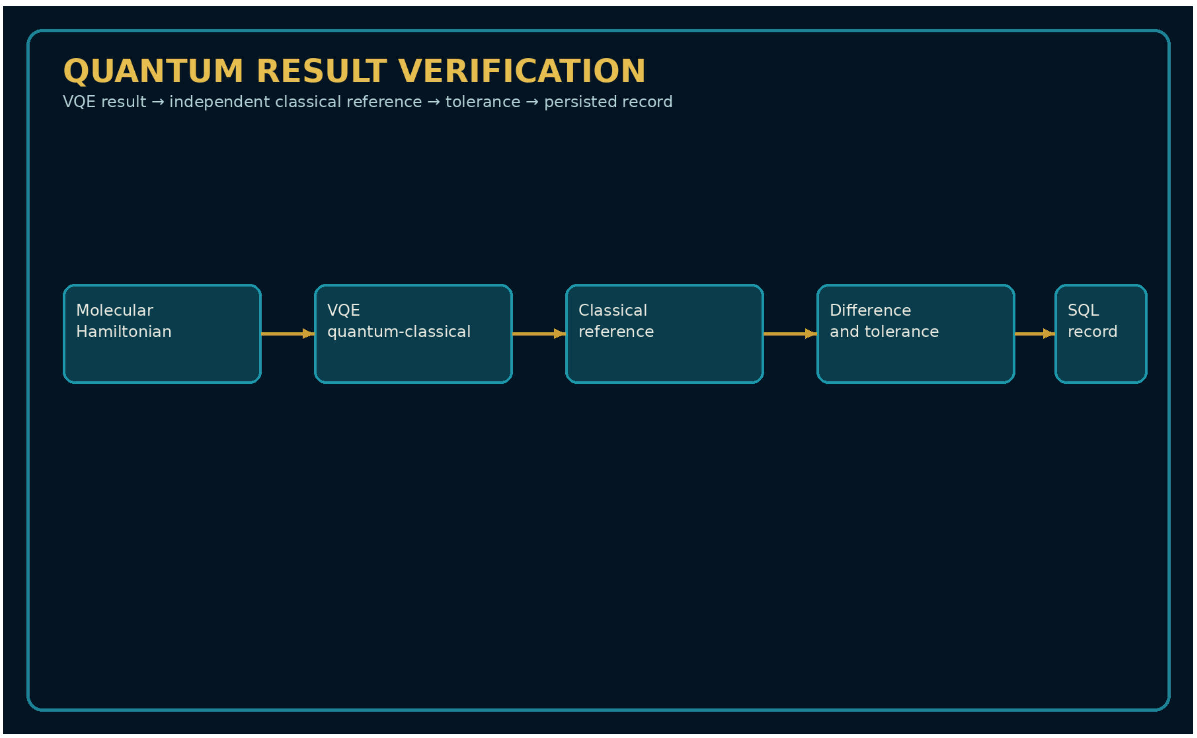
reference = classical_fci_reference()
print(f"Classical FCI total energy: {reference:.10f} Ha")
# Compare this value with the total VQE energy produced by the project.
# The comparison must record geometry, basis, mapper, ansatz, optimizer,
# and software versions before a tolerance is interpreted.
```

نتيجة VQE ليست صحيحة لمجرد أن الخوارزمية انتهت دون خطأ برمجي. في مثال H_2 يمكن إنشاء مرجع كلاسيكي مستقل باستخدام PySCF، ثم مقارنة الطاقة الكلية المحسوبة بالمسار الكمومي ضمن تعريف المسألة نفسه: الهندسة، وقاعدة الأساس، والشحنة، والدوران.

A VQE result is not scientifically validated merely because the program completed without a software error. For H_2, an independent classical reference can be computed with PySCF and compared with the quantum workflow under the same molecular geometry, basis set, charge, and spin.

لوحة: مقارنة النتيجة الكمومية بمرجع كلاسيكي مستقل

Plate: Quantum result verification against an independent classical reference.



12.6 التحقق من النتيجة الكمومية بمرجع كلاسيكي

12.6 Verifying the Quantum Result Against a Classical Reference

SQL

```
SELECT system_name, backend, energy, software_version, created_at
FROM quantum_runs
WHERE system_name = 'H2'
ORDER BY created_at DESC
LIMIT 1;
```

كما يمكن استرجاع أحدث تجربة لنظام معين، وهو أمر مهم عندما تتغير الدائرة أو المحسن أو الإصدار البرمجي.

The latest run for a given system can also be retrieved, which matters when the circuit, optimizer, or software version changes.

SQL

```
SELECT
    system_name,
    backend,
    MIN(energy) AS best_energy,
    software_version
FROM quantum_runs
GROUP BY system_name, backend, software_version
ORDER BY best_energy ASC;
```

تخزين النتائج لا يكتمل دون استعلامات تسمح باسترجاعها ومقارنتها. يمكن لقاعدة البيانات أن تعرض أفضل طاقة لكل نظام وخلفية تشغيل، ثم تربطها بإصدار البرنامج.

Persisting results is incomplete without queries that retrieve and compare them. SQL can expose the best energy for each system and backend and then relate it to the software version used.

13.6 استرجاع النتائج ومقارنتها

13.6 Retrieving and Comparing Quantum Results

المعيار هنا ليس أن يكون خط الأنابيب طويلًا، بل أن تكون كل مرحلة محددة المدخلات والمخرجات وقابلة للفحص.

The goal is not a long pipeline. The goal is a pipeline in which every stage has defined inputs and outputs and can be inspected.

PYTHON / run_pipeline.py

```
from __future__ import annotations
import argparse
import logging
import sqlite3
import subprocess
import sys
from pathlib import Path

ROOT = Path(__file__).resolve().parent
DB = ROOT / "data" / "periodic_table.db"
SCHEMA = ROOT / "scripts" / "build_database.sql"

logging.basicConfig(level=logging.INFO, format="%(asctime)s | %(levelname)s | %(message)s")

def build_database():
    DB.parent.mkdir(parents=True, exist_ok=True)
    with sqlite3.connect(DB) as con:
        con.executescript(SCHEMA.read_text(encoding="utf-8"))

def run(script: str):
    subprocess.run([sys.executable, str(ROOT / script)], check=True)
```

```python
def main():
    parser = argparse.ArgumentParser(description="Brick by Brick Part III pipeline")
    parser.add_argument("--skip-quantum", action="store_true", help="Do not run the optional quantum stage")
    parser.add_argument("--stage", choices=["database","ml","report","all"], default="all")
    args = parser.parse_args()

    if args.stage in ("database", "all"):
        logging.info("Building SQLite schema")
        build_database()
        run("scripts/insert_elements.py")
    if args.stage in ("ml", "all"):
        run("scripts/train_models.py")
    if args.stage in ("report", "all"):
        run("scripts/generate_report.py")
    if args.skip_quantum:
        logging.info("Quantum stage skipped by request")
    else:
        logging.info("Quantum stage is available only when the validated Qiskit/PySCF environment is installed")

if __name__ == "__main__":
    main()
```

بدل تشغيل كل مرحلة يدويًا، يمكن جمع المراحل التي لا تعتمد على خدمات خارجية في سكربت واحد. يظل الجزء الكمومي اختياريًا، ويُسجل فشله أو غياب بيئته بدل أن يخفيه.

Instead of running every stage manually, the stages that do not depend on external services can be coordinated by one script. The quantum stage remains optional, and a missing quantum environment is reported rather than silently ignored.

خط الأنابيب الرئيسي

Main execution pipeline

PYTHON / tests/test_periodic_table.py

```python
import numpy as np
from extract_periodic_table_matrix import extract_numeric_matrix

def test_periodic_table_matrix_shape():
    X, metadata = extract_numeric_matrix("periodic_table.db")
    assert X.shape[0] == 118
    assert X.ndim == 2
    assert len(metadata) == 118

def test_numeric_matrix_has_no_nan_after_imputation():
    X, _ = extract_numeric_matrix("periodic_table.db")
    assert np.isfinite(X).all()
```

تتحول قابلية الاختبار من وعد إلى ممارسة عندما نختبر الدوال التي يعتمد عليها المشروع. المثال التالي يتحقق من عدد العناصر، وشكل المصفوفة، وعدم وجود قيم مفقودة بعد المعالجة.

Testability becomes operational when the project tests the functions on which its workflow depends. The following example checks the number of elements, matrix shape, and absence of missing values after preprocessing.

اختبارات الوحدة

Unit Tests

هذه الملفات لا تعني أن كل جهاز يجب أن يثبت كل الأدوات. يمكن تشغيل الأجزاء الكلاسيكية دون المكونات الكمومية، ثم تثبيت طبقة Qiskit وPySCF عندما يصل التنفيذ إلى الفصل الكمومي.

These files do not mean every machine must install every tool. The classical sections can run without the quantum stack, while Qiskit and PySCF can be installed when the workflow reaches the quantum chapters.

YAML / environment.yml

```
name: brick-by-brick-part-iii
channels:
  - conda-forge
dependencies:
  - python=3.12
  - pip
  - numpy=2.3.5
  - pandas=2.2.3
  - scikit-learn=1.8.0
  - pytest=8.4.1
  - pip:
      - qiskit==2.5.0
      - qiskit-algorithms==0.4.0
      - qiskit-nature==0.8.0
      - pyscf==2.14.0
      - dask==2026.7.1
      - qiskit-ibm-runtime==0.47.0
```

ملف بيئة Conda

Conda environment file

TXT / requirements.txt

```
numpy==2.3.5
pandas==2.2.3
scikit-learn==1.8.0
qiskit==2.5.0
qiskit-algorithms==0.4.0
qiskit-nature==0.8.0
```

```
pyscf==2.14.0
pytest==8.4.1
dask==2026.7.1
# Optional real-hardware client
qiskit-ibm-runtime==0.47.0
```

النسخة الموصى بها هنا مبنية على Python 3.11/3.12، مع تثبيت نسخ واضحة للمكونات الأساسية. حزمة Qiskit Nature 0.8 تتطلب Qiskit حديثًا وqiskit-algorithms بإصدار 0.4 أو أحدث، ولذلك يجب ألا يختلط هذا المسار مع أمثلة Qiskit القديمة.

The recommended environment uses Python 3.11/3.12 with explicit versions for the core components. Qiskit Nature 0.8 requires a modern Qiskit stack and qiskit-algorithms 0.4 or newer, so this workflow should not be mixed with legacy Qiskit examples.

ملف البيئة requirements.txt

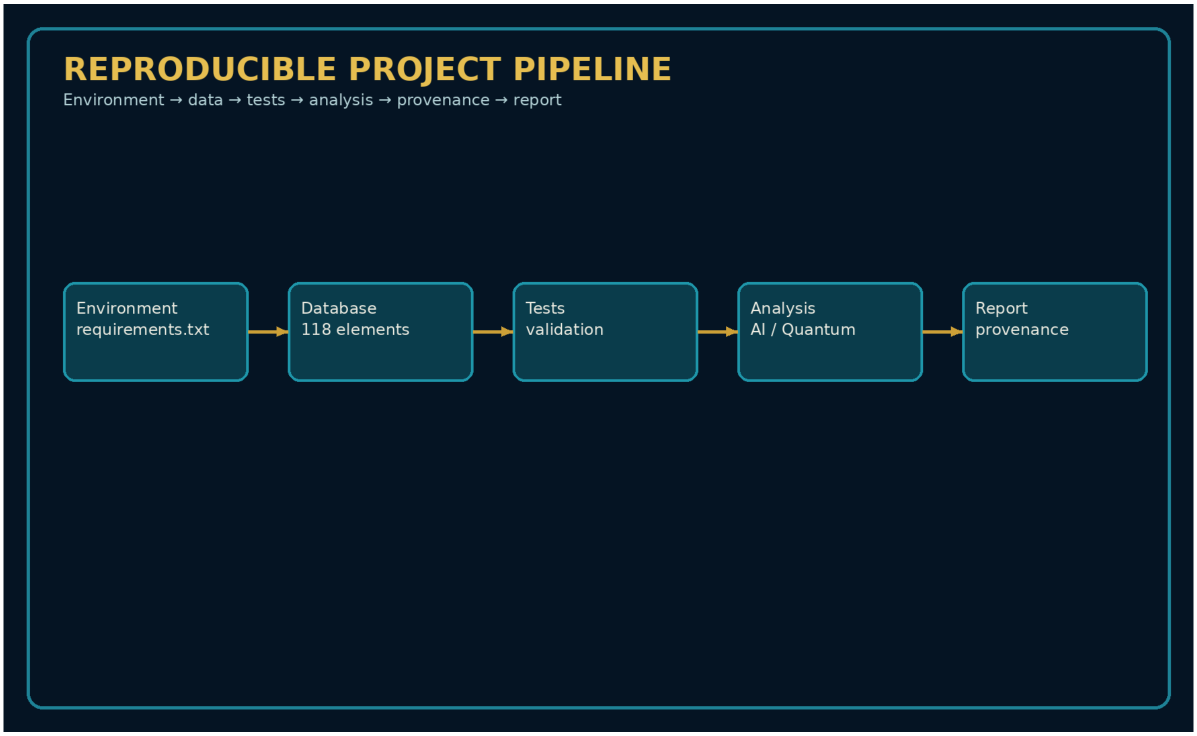
Environment file: requirements.txt

اكتمال المشروع لا يعتمد على وجود الخوارزميات فقط؛ بل على قدرة القارئ على إعادة تشغيلها في بيئة محددة، والتحقق من المخرجات، ومعرفة الإصدار الذي أنتج كل نتيجة. لذلك نضيف ثلاث طبقات تشغيلية: تعريف البيئة، اختبارات الوحدة، وخط أنابيب يجمع المراحل الأساسية.

Project completeness is not only the presence of algorithms. A reader should be able to reproduce the workflow in a defined environment, verify outputs, and identify the software version that produced each result. The project therefore adds three operational layers: environment specification, unit tests, and an execution pipeline that connects the main stages.

لوحة: خط أنابيب قابل لإعادة الإنتاج

Plate: Reproducible execution pipeline



14.8 قابلية إعادة الإنتاج: البيئة والاختبارات وخط الأنابيب

14.6 التوسع من H_2 إلى H_2O: كيف يتغير السؤال الكمومي؟

14.8 Reproducibility: Environment, Tests, and the Execution Pipeline

14.6 Scaling from H_2 to H_2O: How Does the Quantum Question Change?

كان الهيدروجين مناسبًا كبداية لأنه يتيح رؤية الفكرة كاملة بأقل قدر من التعقيد: بيانات كلاسيكية، نموذج إلكتروني، هاملتوني، تمثيل كيوبتي، دائرة، محسن كلاسيكي، ثم نتيجة يمكن حفظها. لكن الانتقال إلى H_2O يوضح للقارئ لماذا لا تعني زيادة عدد الذرات مجرد زيادة عدد الصفوف في قاعدة البيانات. هنا يصبح السؤال الفيزيائي نفسه أكثر تحديدًا: ما الهندسة الجزيئية؟ ما قاعدة الأساس؟ ما الإلكترونات التي ستدخل في النموذج؟ وما التقريب الذي سيجعل المسألة قابلة للمحاكاة؟.

Hydrogen is a useful starting point because it exposes the complete idea with limited complexity: classical data, an electronic-structure model, a Hamiltonian, a qubit representation, a circuit, a classical optimizer, and a result that can be stored. Moving to H_2O shows why adding atoms is not merely equivalent to adding rows to a database. The physical question must become more specific: What molecular geometry is used? What basis set is chosen? Which electrons enter the model? And what approximation makes the calculation tractable.

في هذا المشروع لا ندّعي أن الجدول الدوري وحده يحتوي وصفًا كموميًا للماء. الجدول يزوّدنا بهويات العناصر وخصائصها المرجعية، بينما تأتي البنية الجزيئية والبيانات الإلكترونية من طبقة أخرى. هذه الحدود مهمة لأن النظام الجيد يفصل بين «مصدر العنصر» و«نموذج الجزيء» و«نتيجة المحاكاة».

This project does not claim that the periodic table alone contains a quantum description of water. The table supplies element identities and reference properties, while molecular geometry and electronic-structure data come from another layer. That separation matters because a robust system distinguishes the element source, the molecular model, and the simulation result.

لذلك يمكن تمثيل H_2O داخل المنصة كسجل تجربة له هوية مستقلة: اسم الجزيء، الهندسة، عدد الإلكترونات، النموذج الإلكتروني، قاعدة الأساس، طريقة الاختزال، خريطة الكيوبتات، الخوارزمية، وإصدار البرنامج. وعند تغيير أي عنصر مؤثر في الحساب يجب أن يتغير الإصدار التجريبي بدل أن تُستبدل النتيجة القديمة بصمت.

H_2O can therefore be represented in the platform as an independently identified experiment: molecule name, geometry, electron count, electronic-structure model, basis set, reduction method, qubit mapping, algorithm, and software version. When any calculation-relevant component changes, a new experiment version should be recorded rather than silently replacing the previous result.

مثال بنية سجل تجربة جزيئية:

Example schema for a molecular experiment record:

صندوق الكود — SQL

Code Box — SQL

```
CREATE TABLE IF NOT EXISTS molecular_experiments (
    experiment_id INTEGER PRIMARY KEY,
    molecule TEXT NOT NULL,
    geometry TEXT NOT NULL,
    basis_set TEXT NOT NULL,
    active_space TEXT,
    mapper TEXT NOT NULL,
    algorithm TEXT NOT NULL,
    software_version TEXT NOT NULL,
    result_energy REAL,
    result_unit TEXT,
    status TEXT NOT NULL,
    created_at TEXT NOT NULL
);
```

هذا المثال لا يحسب طاقة الماء بذاته؛ إنه يوضح نقطة هندسية مهمة: النتيجة الكمومية لا تكون ذات قيمة تشغيلية إذا فقدت سياقها. عندما يعود القارئ بعد أشهر إلى سجل الطاقة، يجب أن يعرف لأي نموذج وهندسة وبرنامج تنتمي النتيجة.

This schema does not calculate the energy of water by itself. It illustrates an engineering principle: a quantum result has limited operational value if its context is lost. When a reader returns to an energy record months later, the model, geometry, and software that produced it should still be identifiable.

14.7 CO_2 والهاملتوني: من العناصر إلى نموذج فيزيائي محدد

14.7 CO_2 and the Hamiltonian: From Elements to a Defined Physical Model

يمثل CO_2 مثالًا ثانيًا مفيدًا لأنه يوضح الفرق بين «وجود العناصر» و«وجود نظام فيزيائي». معرفة أن الكربون والأكسجين موجودان في الجدول الدوري لا تحدد وحدها طول الروابط أو الزوايا أو الحالة الإلكترونية أو طاقة النظام. هذه المعلومات يجب أن تأتي من نموذج كيميائي وفيزيائي محدد.

CO_2 provides a second useful example because it separates “element presence” from “physical system definition.” Knowing that carbon and oxygen are present in the periodic table does not by itself determine bond lengths, angles, electronic states, or system energy. Those quantities require a defined chemical and physical model.

في طبقة الحوسبة الكمومية، يتحول النموذج الإلكتروني إلى هاملتوني يمكن كتابته على صورة مؤثرات باولي بعد اختيار طريقة الترميز. عندها تصبح المسألة قابلة للربط بالدائرة الكمومية. لكن عدد حدود الهاملتوني، وعدد الكيوبتات، وعمق الدائرة، وعدد القياسات لا تتحدد من اسم الجزيء وحده؛ بل من الاختيارات الحسابية السابقة.

In the quantum layer, the electronic model is transformed into a Hamiltonian that can be represented with Pauli operators after a qubit-mapping choice. The resulting problem can then be connected to a quantum circuit. The number of Hamiltonian terms, qubits, circuit depth, and measurements is not determined by the molecule name alone; it depends on the preceding computational choices.

وهنا يظهر معنى «الهاملتوني» في المشروع بصورة تعليمية: هو الجسر الرياضي بين الوصف الفيزيائي الذي نريد حسابه وبين العمليات التي تستطيع الخوارزمية الكمومية تنفيذها. إذا تغير الهاملتوني أو الترميز أو التقريب، فليست النتيجة الجديدة مجرد نسخة أسرع من النتيجة القديمة، بل نتيجة لتجربة مختلفة.

This is where the Hamiltonian has a concrete educational role in the project: it is the mathematical bridge between the physical quantity being modeled and the operations that the quantum algorithm can execute. If the Hamiltonian, mapping, or approximation changes, the new result is not merely a faster copy of the old one; it belongs to a different computational experiment.

تمثيل تعليمي مبسط للهاملتوني في صورة مؤثرات باولي:

A simplified educational representation of a Hamiltonian in Pauli operators:

صندوق الكود — PYTHON

Code Box — PYTHON

```
from dataclasses import dataclass

@dataclass(frozen=True)
class PauliTerm:
    coefficient: float
    operator: str

terms = [
    PauliTerm(-0.0421, "IIII"),
    PauliTerm(0.1777, "ZIII"),
    PauliTerm(-0.2427, "ZZII"),
]

hamiltonian = sum(t.coefficient for t in terms)
print("Number of terms:", len(terms))
print("Coefficient sum (diagnostic only):", hamiltonian)
```

هذا الكود تعليمي متعمدًا: هو لا يدّعي أنه هاملتوني CO_2 حقيقي، بل يوضح كيف تصبح الحدود التي تنتجها طبقة الكيمياء الإلكترونية سجلات يمكن فحصها. أما الهاملتوني الفعلي فيُستخرج من نموذج إلكتروني محدد ولا يجوز اختراعه من أرقام توضيحية.

The code is intentionally educational: it does not claim to be a real CO_2 Hamiltonian. It shows how terms produced by an electronic-structure layer can become inspectable records. A real Hamiltonian must be derived from a defined electronic-structure model rather than invented from illustrative numbers.

14.8 دوائر 4–6 كيوبت: ما الذي نتعلمه وما الذي لا نثبته؟

14.8 Four-to-Six-Qubit Circuits: What Do They Teach, and What Do They Not Prove?

الانتقال إلى دوائر من 4–6 كيوبت مفيد لأنه يجعل مشكلة التوسع مرئية للقارئ. عدد الحالات الأساسية لنظام من n كيوبت هو 2^n؛ لذلك تنتقل المساحة من 16 حالة عند أربعة كيوبت إلى 64 حالة عند ستة كيوبت. هذه الزيادة لا تعني تلقائيًا أن الحساب أصبح مفيدًا علميًا، لكنها توضح لماذا تتسع تكلفة التمثيل والمحاكاة بسرعة.

Moving to four-to-six-qubit circuits makes scaling visible. An n-qubit system has 2^n computational basis states, so the state space grows from 16 basis states at four qubits to 64 at six qubits. This growth does not automatically make a calculation scientifically useful, but it makes the rapid expansion of representation and simulation cost concrete.

في التطبيق الحقيقي توجد قيود أخرى: عدد البوابات، عمق الدائرة، الضوضاء، عدد اللقطات، كلفة تقدير القيمة المتوقعة، وجودة المحسن. لذلك يجب تسجيل هذه العوامل إلى جانب الطاقة أو الخاصية النهائية. لا يكفي أن نكتب «VQE = 4 qubits» ثم نعرض رقمًا واحدًا؛ يجب أن نعرف كيف وصلنا إليه.

Real workflows introduce additional constraints: gate count, circuit depth, noise, shot count, expectation-value cost, and optimizer behavior. These should be recorded alongside the final energy or property. It is not enough to write “VQE = 4 qubits” and display one number; the computational path must remain identifiable.

مثال حسابي صغير يوضح نمو فضاء الحالة:

A small calculation illustrating state-space growth:

صندوق الكود — PYTHON

Code Box — PYTHON

```
def basis_state_count(qubits: int) -> int:
    if qubits < 0:
        raise ValueError("qubits must be non-negative")
    return 2 ** qubits

for q in range(1, 7):
    print(f"{q} qubits -> {basis_state_count(q)} basis states")
```

يمكنك استخدام هذا المثال لتفريق مفهومين كثيرًا ما يختلطان: عدد الحالات الرياضية التي يمثلها النظام، وعدد الموارد الفيزيائية أو الحسابية اللازمة للحصول على نتيجة موثوقة. الأول ينمو أُسّيًا مع عدد الكيوبتات، بينما الثاني يعتمد أيضًا على الخوارزمية والجهاز والضوضاء وطريقة القياس.

This example separates two ideas that are often conflated: the number of mathematical basis states represented by a system and the physical or computational resources required for a reliable result. The former grows exponentially with qubit count, while the latter also depends on the algorithm, hardware, noise, and measurement strategy.

14.9 HHL: أين يدخل حل الأنظمة الخطية؟

14.9 HHL: Where Does Quantum Linear-System Solving Enter?

خوارزمية HHL مثال مهم على ضرورة الدقة في وصف الخوارزميات الكمومية. هي مصممة لحل أنظمة خطية من الشكل A x = b تحت افتراضات وشروط محددة. لذلك لا يصح تقديمها على أنها خوارزمية جاهزة «لحل معادلة شرودنجر» بمجرد إعطائها اسم المعادلة. في تطبيق فيزيائي يجب أولًا تحويل المسألة إلى نظام خطي مناسب، ثم التأكد من خصائص المصفوفة وطريقة تجهيز الحالة واستخراج الكمية المطلوبة.

HHL is an important example of why quantum algorithms must be described precisely. It is designed for linear systems of the form A x = b under specific assumptions and conditions. It should therefore not be presented as a ready-made “Schrödinger-equation solver” simply by naming the equation. A physical application first requires a suitable linear-system formulation, followed by attention to matrix properties, state preparation, and extraction of the desired quantity.

ومن أهم القيود التعليمية التي ينبغي أن تبقى أمام القارئ: حالة المصفوفة، عدد الشرط، كلفة تحميل البيانات إلى الحالة الكمومية، وما الذي يمكن استخراجه بكفاءة من الناتج. قد تكون الخوارزمية ذات تعقيد نظري جذاب، لكن ذلك لا يضمن تفوقًا عمليًا على خوارزمية كلاسيكية في كل حالة.

Key educational constraints include matrix conditioning, state-preparation cost, data-loading overhead, and what information can be extracted efficiently from the quantum state. A theoretically attractive complexity does not guarantee a practical advantage over a classical algorithm in every problem.

دون ادعاء تنفيذ في هذا المشروع هيكل تعليمي لمسار HHL:

An educational HHL workflow without claiming an implementation in this project:

صندوق الكود — PYTHON

Code Box — PYTHON

```
def hhl_workflow_check(matrix_name, rhs_name, condition_number, data_loading_cost):
    checks = {
        "linear_system_defined": bool(matrix_name and rhs_name),
        "conditioning_recorded": condition_number is not None,
        "data_loading_recorded": data_loading_cost is not None,
    }
    return checks

print(hhl_workflow_check("A", "b", condition_number=12.4,
                         data_loading_cost="documented"))
```

هذا الكود ليس HHL ولا يحل نظامًا خطيًا؛ وظيفته أن يوضح كيف يمكن للنظام أن يرفض وصف تجربة بأنها مكتملة إذا كانت مدخلاتها وقيودها غير موثقة. هذه هي الفكرة الهندسية التي نحتاجها قبل إدخال خوارزمية متقدمة.

This code is not HHL and does not solve a linear system. Its purpose is to show how a system can refuse to treat an experiment as complete when its inputs and constraints are undocumented. That engineering discipline should precede the introduction of an advanced algorithm.

دون مبالغة LLMsو VAEsو GANs :الذكاء الاصطناعي التوليدي 14.10

14.10 Generative AI: GANs, VAEs, and LLMs Without Overclaiming

يمكن أن تتسع طبقة الذكاء الاصطناعي من التنبؤ والتصنيف إلى التوليد، لكن التوليد هنا يعني «اقتراح مرشحين» وليس «إثبات أن المرشح صالح». يمكن لـ VAE أن يتعلم تمثيلًا كامنًا لبيانات منظمة، ويمكن لـ GAN أن يتعلم توليد عينات تشبه توزيع التدريب، ويمكن لنموذج لغوي أن يعيد تركيب معلومات أو يقترح أوصافًا نصية. في كل حالة يجب أن تبقى مرحلة التحقق مستقلة عن مرحلة التوليد.

The AI layer can expand from prediction and classification to generation, but generation means “proposing candidates,” not “proving that a candidate is valid.” A VAE can learn a latent representation of structured data, a GAN can learn to generate samples resembling its training distribution, and a language model can reorganize information or propose textual descriptions. In every case, validation must remain independent of generation.

لا توجد 118 عينة كافية لتدريب VAE أو GAN في مثال الجدول الدوري، لذلك ذي قيمة علمية عامة. يجب أن تأتي بيانات التدريب من مجموعة موثقة أكبر، بينما يبقى الجدول الدوري طبقة مرجعية للهوية والخصائص الأساسية. هذه النقطة تمنع خطأ شائعًا: مساواة «وجود جدول صغير» مع «وجود مجموعة تدريب كافية».

The 118-element table is not enough to train a generally meaningful GAN or VAE. A real training dataset would need to come from a larger documented collection, while the periodic table remains a reference layer for identities and core properties. This prevents a common mistake: treating the existence of a small table as equivalent to having a sufficient training corpus.

مثال مبسط لطبقة توليد مرشحين باستخدام VAE على ميزات عددية:

A simplified VAE-style candidate-generation layer over numerical features:

صندوق الكود — **PYTHON**

Code Box — PYTHON

```
# Educational PyTorch skeleton; training requires a documented dataset X.
import torch
import torch.nn as nn

class Encoder(nn.Module):
    def __init__(self, n_features, latent_dim=4):
        super().__init__()
        self.net = nn.Sequential(nn.Linear(n_features, 32), nn.ReLU())
        self.mu = nn.Linear(32, latent_dim)
        self.logvar = nn.Linear(32, latent_dim)

    def forward(self, x):
        h = self.net(x)
        return self.mu(h), self.logvar(h)

class Decoder(nn.Module):
    def __init__(self, latent_dim=4, n_features=8):
        super().__init__()
        self.net = nn.Sequential(nn.Linear(latent_dim, 32), nn.ReLU(),
                                 nn.Linear(32, n_features))

    def forward(self, z):
        return self.net(z)
```

هذا المقتطف يوضح بنية التشفير وفك التشفير، لكنه لا يقدم نموذجًا مدربًا ولا يثبت أن أي مادة جديدة آمنة أو قابلة للتصنيع. قبل استخدامه في مشروع علمي يجب تثبيت مصدر البيانات، وتقسيم التدريب والاختبار، ومعيار التقييم، ونسخة النموذج، وآلية فحص المرشحين بعد التوليد.

The snippet shows encoder and decoder structure, but it is not a trained model and does not establish that any generated material is safe or synthesizable. Before scientific use, the dataset source, train/test split, evaluation metric, model version, and post-generation candidate checks must be recorded.

أما LLMs فيمكن أن تكون مفيدة في طبقة التفسير، وتلخيص السجلات، واستخراج المعلومات المنظمة، واقتراح أسئلة بحثية أو أوصاف أولية. لكنها لا ينبغي أن تكون المصدر الوحيد لقيمة فيزيائية أو كيميائية، ولا يجوز أن تحل محل قاعدة بيانات موثقة أو حساب مستقل.

LLMs can be useful for interpretation, record summarization, structured-information extraction, research-question generation, and preliminary descriptions. They should not be the sole source of a physical or chemical value, and they should not replace a documented database or an independent calculation.

نظام توصية قابل للتدقيق: من الهدف إلى المرشح 14.11

14.11 An Auditable Recommendation System: From Target to Candidate

أفضل نقطة بداية في مشروع صغير هي نظام توصية واضح بدل نظام توليدي غامض. تحدد أولًا متجهًا للخواص المطلوبة، ثم تطبع الميزات على مقياس مشترك، ثم تحسب المسافة أو التشابه، ثم تعرض أفضل المرشحين مع سبب اختيارهم. إذا تغيرت الأوزان أو الميزات، يجب أن يتغير إصدار التوصية.

A transparent recommendation system is often a better starting point for a small project than an opaque generative system. First define a target property vector, scale features onto a common basis, compute a distance or similarity, and return candidates together with the reason for ranking them. If weights or features change, the recommendation version should change as well.

مثال قابل لإعادة التشغيل على أقرب الجيران:

A reproducible nearest-neighbor example:

صندوق الكود — PYTHON

Code Box — PYTHON

```
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import NearestNeighbors

# Rows are documented element-feature records.
X = np.asarray(X, dtype=float)
target = np.asarray(target, dtype=float).reshape(1, -1)

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
target_scaled = scaler.transform(target)

model = NearestNeighbors(n_neighbors=5, metric="euclidean")
model.fit(X_scaled)
distance, index = model.kneighbors(target_scaled)

for d, i in zip(distance[0], index[0]):
    print(i, float(d))
```

القيمة التعليمية هنا هي قابلية التفسير: يستطيع القارئ أن يعرف ما الذي قورن وما المقياس المستخدم. لكن لا يجوز تحويل «أقرب مرشح عدديًا» إلى حكم بأنه أفضل مادة؛ التوصية مرحلة فرز، بينما الحكم العلمي يحتاج خصائص إضافية وحسابًا مستقلًا وربما تحققًا تجريبيًا.

The educational value is interpretability: the reader can see what was compared and which metric was used. A numerically nearest candidate must not be converted into a claim that it is the best material. Recommendation is a screening stage; scientific judgment requires additional properties, independent calculations, and possibly experimental validation.

14.12 الجسر الجينومي: Gene Resistance Tracker كطبقة بيانات

14.12 The Genomic Bridge: Gene Resistance Tracker as a Data Layer

يمكن أن يصبح Gene Resistance Tracker جسرًا بين هذا المشروع ومشكلات البيانات الجينومية، بشرط ألا نخلط بين المثال التعليمي والبيانات البيولوجية الحقيقية. في نظام فعلي قد نحتاج إلى ربط معرف الجين، المتغير، العينة، النمط الظاهري، مصدر الدليل، إصدار قاعدة البيانات، ودرجة الثقة. هذه الروابط تجعل النتيجة قابلة للمراجعة وتمنع فقدان أصل المعلومة.

Gene Resistance Tracker can serve as a bridge to genomic-data problems, provided that the educational example is not confused with real biological data. A real system may need to link gene identifiers, variants, samples, phenotypes, evidence sources, database versions, and confidence levels. Such links preserve provenance and make results reviewable.

نموذج بيانات تعليمي للمقاومة:

An educational data model for resistance observations:

صندوق الكود — SQL

Code Box — SQL

```
CREATE TABLE IF NOT EXISTS resistance_observations (
    observation_id INTEGER PRIMARY KEY,
    gene_id TEXT NOT NULL,
    variant_id TEXT,
    phenotype TEXT NOT NULL,
    evidence_source TEXT NOT NULL,
    evidence_level TEXT NOT NULL,
    dataset_version TEXT NOT NULL,
    confidence REAL CHECK (confidence >= 0.0 AND confidence <= 1.0)
);

CREATE INDEX IF NOT EXISTS idx_resistance_gene
ON resistance_observations(gene_id);
```

لا تحتوي هذه الجداول على مرضى ولا نتائج سريرية. وهي لا تقدم بروتوكولًا لتعديل الجينات أو مقاومة دواء؛ وظيفتها أن تشرح كيف تُبنى طبقة بيانات قابلة للتتبع عندما نصل لاحقًا إلى بيانات جينومية مصرح بها. في المشروع الحالي يبقى الجدول الدوري مثالًا صغيرًا يسمح لك بتعلم البنية قبل التعامل مع الحجم الحقيقي.

These tables contain no patients or clinical outcomes. They do not provide a protocol for gene editing or drug resistance; their role is to show how a traceable data layer can be built before authorized genomic datasets are introduced. In the present project, the periodic table remains the small controlled example that lets you learn the architecture before facing real scale.

وعند ربط العناصر بالبروتينات والإنزيمات يجب أيضًا التمييز بين الارتباط الكيميائي والسببية البيولوجية. وجود عنصر في بنية جزيئية لا يثبت أنه سبب مرض أو مقاومة. السبب يحتاج إلى تصميم دراسة مناسب وأدلة مستقلة. لذلك تبقى قاعدة البيانات هنا وسيلة لتنظيم الأدلة، لا آلة لإنتاج سببية من الارتباط.

When linking elements to proteins or enzymes, chemical association must also be separated from biological causality. The presence of an element in a molecular structure does not prove that it causes disease or resistance. Causality requires an appropriate study design and independent evidence. The database therefore organizes evidence rather than manufacturing causality from association.

14.13 التوسع إلى Dask وSpark: متى يصبح الحجم مشكلة حقيقية؟

14.13 Scaling to Dask and Spark: When Does Size Become a Real Problem?

الجدول الدوري صغير عمدًا، ولذلك لا يحتاج إلى بنية موزعة. هذه نقطة تعليمية مهمة: استخدام Spark أو Dask هنا لمجرد إظهار أداة كبيرة سيكون عكس هندسة النظام الجيدة. تبدأ الحاجة عندما تتجاوز البيانات الذاكرة المتاحة، أو عندما تصبح العمليات المتكررة مكلفة، أو عندما تكون هناك ملفات كثيرة يمكن معالجتها بالتوازي.

The periodic table is intentionally small, so it does not require distributed infrastructure. This is an important engineering lesson: introducing Spark or Dask merely to display a large tool would be the opposite of good system design. The need appears when data exceed available memory, repeated operations become expensive, or many files can be processed in parallel.

مثال Dask تعليمي على بيانات كبيرة مقسمة إلى أجزاء:

An educational Dask example over partitioned large data:

صندوق الكود — PYTHON

Code Box — PYTHON

```
import dask.dataframe as dd

# Example: many partitioned CSV files, not the 118-row periodic table.
df = dd.read_csv("data/large_features/*.csv", assume_missing=True)
summary = df.groupby("element")[["feature_1", "feature_2"]].mean().compute()
print(summary.head())
```

هنا يمثل مرحلة يمكن الوصول إليها عندما Dask الفرق بين هذا المثال والجدول الدوري هو المقصود: تتغير طبيعة البيانات، لا خطوة يجب فرضها على كل نسخة من المشروع. وفي مشروع جينومي حقيقي يمكن أن يكون هذا الانتقال أكثر أهمية بسبب تعدد الملفات والأحجام الكبيرة وعمليات القراءة والتحويل المتكررة.

The distinction between this example and the periodic table is intentional: Dask represents a stage that becomes useful when data characteristics change, not a mandatory step for every version of the project. In a real genomic project, the transition can become much more relevant because of many large files and repeated read/transform operations.

14.14 PyTorch وTensorFlow: متى يستحق التعلم العميق مكانه؟

14.14 PyTorch and TensorFlow: When Does Deep Learning Earn Its Place?

التعلم العميق مناسب عندما تكون هناك بيانات كافية وبنية مشكلة تبرر نموذجًا عميقًا، مثل بيانات صور أو تسلسلات أو تمثيلات عالية الأبعاد. لا يضيف TensorFlow أو PyTorch قيمة تلقائية إلى جدول صغير. في هذا المشروع يمكن تقديمهما كامتداد اختياري، مع الحفاظ على نموذج scikit-learn الأبسط كنقطة مرجعية يمكن مقارنة النموذج العميق بها.

Deep learning is justified when enough data and a problem structure support a deep model, such as images, sequences, or high-dimensional representations. PyTorch or TensorFlow do not automatically add value to a small table. In this project they are optional extensions, while the simpler scikit-learn model remains a reference baseline against which a deep model can be compared.

قاعدة تحقق بسيطة قبل تدريب نموذج أعمق:

A simple gate before training a deeper model:

صندوق الكود — PYTHON

Code Box — PYTHON

```python
def justify_deep_learning(n_samples, n_features, baseline_score, target_score):
    return {
        "enough_samples": n_samples >= 1000,
        "feature_space_nontrivial": n_features >= 20,
        "baseline_is_insufficient": target_score > baseline_score,
    }

print(justify_deep_learning(5000, 64, 0.71, 0.82))
```

هذه ليست قاعدة علمية عامة ولا حدًا رسميًا لبدء التعلم العميق؛ إنها بوابة تعليمية تذكّر بأن قرار اختيار النموذج يجب أن يستند إلى حجم البيانات وأداء خط الأساس وطبيعة المهمة. الحدود الفعلية يجب أن يحددها تصميم التجربة وخصائص البيانات.

This is not a universal scientific rule or a formal threshold for starting deep learning. It is an educational gate that emphasizes that model choice should consider data size, baseline performance, and task structure. Actual thresholds must be determined by the experimental design and dataset.

14.15 الانتقال إلى Qiskit Runtime وAWS Braket: سجل العتاد قبل الرقم

14.15 Moving to Qiskit Runtime and AWS Braket: Record the Hardware Before the Number

عندما تنتقل من محاكاة محلية إلى خدمة كمومية أو جهاز حقيقي، تتغير طبيعة السجل التجريبي. يجب حفظ اسم الجهاز أو الخدمة، إصدار البرنامج، الدائرة بعد عملية transpilation، عدد القياسات، وقت التنفيذ، وأي تقنية لتخفيف الخطأ. هذه المعلومات ليست تفاصيل جانبية؛ فهي جزء من تفسير النتيجة.

When moving from a local simulation to a quantum service or real device, the experimental record changes. The device or service name, software version, transpiled circuit, shot count, execution time, and any error-mitigation technique should be stored. These are not peripheral details; they are part of interpreting the result.

مثال سجل تشغيل كمومي قابل للحفظ في SQL:

An executable record structure for a quantum run:

صندوق الكود — SQL

Code Box — SQL

```
CREATE TABLE IF NOT EXISTS quantum_runs (
    run_id INTEGER PRIMARY KEY,
    experiment_id INTEGER NOT NULL,
    backend TEXT NOT NULL,
    software_version TEXT NOT NULL,
    circuit_hash TEXT NOT NULL,
    shots INTEGER NOT NULL,
    mitigation TEXT,
    energy REAL,
    unit TEXT,
    status TEXT NOT NULL,
    FOREIGN KEY (experiment_id) REFERENCES molecular_experiments(experiment_id)
);
```

هذا backend يمكن أن يكون محاكيًا محليًا أو خدمة سحابية أو جهازًا حقيقيًا. المهم ألا تخفي المنصة الفرق. نتيجة محاكاة مثالية ليست مساوية لنتيجة جهاز حقيقي، حتى لو كان اسم الخوارزمية واحدًا؛ لأن الضوضاء والقياس والدوائر المنفذة تغير ظروف التجربة.

The backend may be a local simulator, a cloud service, or real hardware. The platform should make that distinction explicit. An ideal simulation result is not equivalent to a hardware result even when the algorithm name is the same, because noise, measurement, and executed circuits change the experimental conditions.

14.16 الموافقة المسبقة والحوكمة: من المبدأ إلى بوابة تشغيلية

14.16 Pre-Approval and Governance: From Principle to an Operational Gate

كلما اتسعت طبقات التوليد والمحاكاة، زاد عدد النتائج التي يمكن إنتاجها قبل أن يراجعها إنسان. لذلك يمكن تحويل الحوكمة إلى بوابة تشغيلية: تعريف الهدف، تحديد البيانات المسموح بها، تسجيل المستخدم والنسخة، فحص المخاطر، ثم السماح بالتنفيذ أو رفضه. لا تعني الموافقة المسبقة أن كل نتيجة صحيحة؛ إنها تعني أن المسار نفسه مرّ من نقطة تحقق معروفة.

As generative and simulation layers expand, the number of outputs can grow faster than human review. Governance can therefore become an operational gate: define the objective, identify permitted data, record the user and version, perform a risk check, and then allow or reject execution. Pre-approval does not make a result correct; it means the workflow passed a known control point.

نموذج بوابة موافقة مسبقة تعليمية:

An educational pre-approval gate:

صندوق الكود — PYTHON

Code Box — PYTHON

```
from dataclasses import dataclass

@dataclass(frozen=True)
class ApprovalRequest:
    purpose: str
    dataset_version: str
    model_version: str
    risk_level: str
    reviewer: str | None = None


def approve(req: ApprovalRequest) -> bool:
    if not req.purpose or not req.dataset_version or not req.model_version:
        return False
    if req.risk_level == "high" and not req.reviewer:
        return False
    return True

request = ApprovalRequest(
    purpose="educational candidate ranking",
    dataset_version="periodic-table-v1",
    model_version="baseline-nn-v1",
    risk_level="low",
)
print(approve(request))
```

هذه البوابة ليست نظامًا تنظيميًا قانونيًا، ولا تزعم أنها تكفي لاتخاذ قرار سريري أو أمني. إنها مثال هندسي على فكرة أن القيود الأخلاقية يمكن أن تصبح شروطًا قابلة للفحص داخل البرنامج. عندما ترتفع المخاطر، يجب أن ترتفع صرامة المراجعة والأدلة المطلوبة.

This gate is not a legal governance system and is not sufficient for clinical or security decisions. It is an engineering example showing how ethical constraints can become checkable program conditions. As risk increases, the rigor of review and evidence requirements should increase as well.

في هذا الجزء تُفهم الحوكمة بوصفها حماية للمستخدم والبيانات والنتائج من الاستخدام غير الموثق أو المتسرع. ليست الغاية إيقاف الحساب، بل وضعه داخل مسار يمكن تفسيره ومراجعته وإيقافه عندما لا تتوافر شروط الثقة.

In this part, governance is understood as protection for users, data, and results against undocumented or premature use. The goal is not to stop computation, but to place it inside a workflow that can be explained, audited, and halted when trust conditions are not met.

14.17 معيار الإغلاق العلمي للفصل: ماذا يجب أن تستطيع إثباته؟

14.17 Scientific Closure Criterion: What Should You Be Able to Demonstrate?

قبل الانتقال إلى بقية المشروع، ينبغي أن تستطيع الإجابة عن خمسة أسئلة: ما البيانات التي دخلت؟ ما التحويلات التي حدثت؟ ما الذي حُسب فعليًا؟ ما الذي تم اقتراحه فقط؟ وما الذي لا يمكن استنتاجه من النتيجة؟ هذه الأسئلة تمنع أن يتحول اتساع التقنيات إلى اتساع غير منضبط في الادعاءات.

Before moving on, you should be able to answer five questions: What data entered? What transformations occurred? What was actually computed? What was only proposed? And what cannot be inferred from the result? These questions prevent technological breadth from becoming uncontrolled breadth of claims.

1. الجدول الدوري مثال صغير ومنظم لتعلم هندسة البيانات، وليس جينومًا كاملًا.

1. The periodic table is a small, structured example for data engineering; it is not a complete genome.

2. الهيدروجين هو مثال كمومي تعليمي أساسي، بينما H_2O وCO_2 يوضحان مشكلة التوسع والنمذجة.

2. Hydrogen is the basic quantum teaching example, while H_2O and CO_2 expose scaling and modeling issues.

3. VQE خوارزمية هجينة تحتاج إلى نموذج فيزيائي محدد وقياسات ومحسن كلاسيكي.

3. VQE is a hybrid algorithm that requires a defined physical model, measurements, and a classical optimizer.

4. HHL خوارزمية للأنظمة الخطية تحت افتراضات محددة، وليست حلًا مباشرًا عامًا لمعادلة شرودنجر.

4. HHL solves linear systems under specific assumptions; it is not a general direct solver for the Schrödinger equation.

5. GANs وVAEs وLLMs تولد أو تقترح مخرجات، لكنها لا تستبدل التحقق العلمي المستقل.

5. GANs, VAEs, and LLMs generate or propose outputs, but they do not replace independent scientific validation.

6. Gene Resistance Tracker طبقة بيانات تعليمية تربط الهوية والمصدر والإصدار والدليل، ولا يستنتج السببية من الارتباط.

6. Gene Resistance Tracker is an educational data layer for identity, provenance, versioning, and evidence; it does not infer causality from association.

7. Dask وSpark أدوات للتوسع عندما يبرر حجم البيانات أو كلفة الحساب استخدامها.

7. Dask and Spark are scaling tools to introduce when data volume or computational cost justifies them.

8. Qiskit Runtime وAWS Braket يضيفان طبقة عتاد وخدمة يجب أن تظهر في سجل التجربة.

8. Qiskit Runtime and AWS Braket add a hardware/service layer that should appear in the experiment record.

9. الموافقة المسبقة والحوكمة يمكن تمثيلهما كحواجز تشغيلية قابلة للفحص وليستا شعارات منفصلة عن النظام.

9. Pre-approval and governance can be implemented as checkable operational gates rather than slogans outside the system.

بهذا يصبح التوسع العلمي امتدادًا طبيعيًا لما بنيته قبل ذلك: البيانات تُعرّف، ثم تُخزّن، ثم تُحوّل إلى تمثيلات عددية، ثم تُحلل، ثم يمكن أن تدخل طبقة تعلم آلي أو توليد أو تمثيل كمومي. وفي كل انتقال يبقى السؤال نفسه: ما الذي تعرفه البيانات؟ وما الذي يثبته الحساب؟ وما الذي يحتاج إلى دليل مستقل؟.

The scientific expansion therefore remains a natural continuation of what came before: data are defined, stored, transformed into numerical representations, analyzed, and then optionally passed to machine learning, generation, or quantum representation. At every transition, the same questions remain: What does the data actually contain? What does the computation establish? And what requires independent evidence.

14.18 الموثوقية العددية: الرقم ليس حقيقة بمجرد ظهوره

14.18 Numerical Reliability: A Number Is Not a Fact Merely Because It Appears

عندما ينتقل المشروع من SQL إلى المصفوفات ثم إلى التعلم الآلي والحوسبة الكمومية، تصبح الموثوقية العددية طبقة مستقلة. قيمة الطاقة أو المسافة أو الدقة أو احتمال التصنيف يجب أن تكون مصحوبة بوحدة القياس، وطريقة الحساب، والتقريب العددي، وأحيانًا بخطأ أو مجال عدم يقين. وإلا يصبح الرقم منفصلًا عن شروط إنتاجه.

When the project moves from SQL to matrices, machine learning, and quantum computing, numerical reliability becomes its own layer. An energy, distance, accuracy, or classification probability should be accompanied by units, calculation method, numerical precision, and sometimes an error or uncertainty range. Otherwise the number becomes detached from the conditions that produced it.

من المفيد أيضًا التمييز بين خطأ النموذج وخطأ التنفيذ. قد يكون النموذج الفيزيائي نفسه تقريبيًا، أو قد يكون البرنامج صحيحًا لكن البيانات تحتوي على قيم مفقودة أو مقاييس مختلفة. لذلك ينبغي أن تسأل قبل مقارنة نتيجتين: هل استخدمتا نفس البيانات؟ نفس الوحدات؟ نفس التحويل؟ نفس إصدار البرنامج؟.

It is also useful to distinguish model error from implementation error. The physical model may be approximate, or the software may be correct while the data contain missing values or incompatible scales. Before comparing two results, ask whether they used the same data, units, transformation, and software version.

صندوق الكود — PYTHON

Code Box — PYTHON

```
import numpy as np

def relative_error(reference, estimate):
    reference = float(reference)
    estimate = float(estimate)
    if reference == 0:
        raise ValueError("reference must be non-zero")
    return abs(estimate - reference) / abs(reference)

ref = -1.137270
est = -1.137100
print(f"relative error = {relative_error(ref, est):.3e}")
```

هذا المثال يحسب خطأً نسبيًا بسيطًا ولا يقرر وحده أن النتيجة مقبولة. معيار القبول يجب أن يأتي من طبيعة المسألة، ودقة المرجع، والغرض من الاستخدام. في الكيمياء الكمومية قد تكون فروق صغيرة مهمة، بينما في مهمة استكشافية قد يكون الهدف مجرد ترتيب المرشحين.

This example computes a simple relative error; it does not decide by itself whether a result is acceptable. Acceptance criteria must come from the problem, reference accuracy, and intended use. In quantum chemistry, small differences can matter, while an exploratory ranking task may only require stable ordering of candidates.

14.19 الجبر الخطي كطبقة وصل بين SQL والذكاء الاصطناعي والكم

14.19 Linear Algebra as the Bridge Between SQL, AI, and Quantum Computing

عند استخراج البيانات من SQL إلى مصفوفة، أنت لا تقوم بمجرد تغيير شكل التخزين. أنت تنقل البيانات إلى تمثيل يسمح بعمليات مثل الضرب المصفوفي، التحليل الطيفي، الإسقاط، وحساب المسافات. هذه العمليات هي اللغة المشتركة بين كثير من خوارزميات التعلم الآلي وبين الصياغة الرياضية للمشكلات الكمومية.

When data are extracted from SQL into a matrix, you are not merely changing storage format. You are moving the data into a representation that supports matrix multiplication, spectral analysis, projection, and distance calculations. These operations form a common language shared by many machine-learning algorithms and the mathematical formulation of quantum problems.

لهذا السبب ينبغي ألا يُعرض الجبر الخطي في المشروع كفصل منفصل عن بقية الطبقات. المتجه الذي يمثل خصائص عنصر يمكن أن يدخل PCA، ثم نموذج تصنيف، ثم يُعاد استخدامه كمدخل لترميز حالة كمومية. المهم هو توثيق التحويلات حتى لا يبدو أن كل طبقة تستخدم «نفس البيانات» بينما هي في الحقيقة تستخدم تمثيلًا مختلفًا لها.

For this reason, linear algebra should not be treated as a chapter disconnected from the other layers. A vector representing element properties can enter PCA, then a classifier, and later be reused as input to a quantum-state encoding. The transformations must be documented so that readers do not assume every layer uses “the same data” when it actually uses a different representation.

صندوق الكود — PYTHON

Code Box — PYTHON

```
import numpy as np

# A tiny, auditable feature matrix: rows are records, columns are features.
X = np.array([[1.0, 2.0, 3.0],
              [2.0, 1.0, 4.0],
              [4.0, 3.0, 2.0]])

# Centering is a common first step before PCA.
X_centered = X - X.mean(axis=0, keepdims=True)
U, S, Vt = np.linalg.svd(X_centered, full_matrices=False)
print("singular values:", S)
print("first principal direction:", Vt[0])
```

نتيجة PCA أو SVD لا تكفي وحدها لإثبات تفسير علمي للعنصر. الاتجاهات الرياضية تعتمد على الميزات التي اخترتها، ولذلك فإن اختيار الميزات جزء من الفرضية نفسها. إذا تغيرت الميزات، قد تتغير المكونات الرئيسية، ويجب تسجيل ذلك في إصدار التحليل.

An SVD or PCA result alone does not establish a scientific interpretation of an element. Mathematical directions depend on the selected features, so feature selection is part of the hypothesis itself. If the features change, principal components may change, and the analysis version should record that change.

الحوسبة الهجينة: لماذا لا يكفي وصف النظام بأنه «كمومي»؟ 14.20

14.20 Hybrid Computing: Why Calling the System “Quantum” Is Not Enough

المشروع هجين بطبيعته. قاعدة البيانات ومعالجة الملفات والتحقق والرسوم والتقارير كلها يمكن أن تعمل كلاسيكيًا، بينما تستخدم طبقة الكم فقط عندما توجد مسألة مناسبة. حتى VQE نفسه حلقة هجينة: المحسن كلاسيكي، والدائرة كمومية، وقياس الطاقة يعاد إلى المحسن. لذلك فإن وصف النظام بأنه «كمومي» يخفي جزءًا كبيرًا من بنيته الفعلية.

The project is inherently hybrid. Database operations, file processing, validation, visualization, and reporting can remain classical, while the quantum layer is used only when an appropriate problem exists. VQE itself is hybrid: the optimizer is classical, the circuit is quantum, and the measured energy returns to the optimizer. Calling the whole system “quantum” therefore hides much of its actual architecture.

يمكنك تصور البنية كخمس طبقات: تخزين البيانات، التحليل العددي، نموذج الذكاء الاصطناعي، التنفيذ الكمومي، ثم طبقة التحقق والحوكمة. تنتقل البيانات بين الطبقات عبر عقود واضحة للمدخلات والمخرجات. إذا فشلت طبقة الكم، لا ينبغي أن يختفي السجل؛ بل يجب أن تسجل المنصة أن المرحلة لم تنفذ أو أنها فشلت، مع السبب.

You can view the architecture as five layers: data storage, numerical analysis, AI modeling, quantum execution, and verification/governance. Data move between layers through explicit input/output contracts. If the quantum layer fails, the record should not disappear; the platform should record that the stage was skipped or failed, together with the reason.

صندوق الكود — PYTHON

Code Box — PYTHON

```python
from dataclasses import dataclass

@dataclass
class StageResult:
    stage: str
    status: str
    message: str


def run_optional_quantum(enabled: bool) -> StageResult:
    if not enabled:
        return StageResult("quantum", "skipped", "quantum environment not enabled")
    # A real implementation would call the validated quantum backend here.
    return StageResult("quantum", "ready", "backend selected and validated")

print(run_optional_quantum(False))
```

الفائدة هنا أن فشل المكون الكمومي لا يتحول إلى فشل صامت أو إلى رقم مصطنع. هذا النمط مهم في الأنظمة العلمية: «لم يُحسب» معلومة أفضل من رقم لا نعرف مصدره.

The benefit is that a missing quantum component does not become a silent failure or an invented number. In scientific systems, “not computed” is often more informative than a number whose origin is unknown.

14.21 المقارنة المرجعية Benchmark: كيف تعرف أن الإضافة تستحق التعقيد؟

14.21 Benchmarking: How Do You Know an Added Layer Is Worth the Complexity?

قبل إدخال أداة جديدة، أنشئ خط أساس. إذا كان نموذج scikit-learn البسيط يعطي أداءً مستقرًا، فانتقالك إلى شبكة عميقة يجب أن يبرر نفسه بقياس واضح. وإذا كان الاستعلام في SQLite يستغرق أجزاء من الثانية، فإدخال Spark لن يكون تحسينًا تلقائيًا. وإذا كانت المحاكاة الكلاسيكية تحل المثال بسرعة وبدقة كافية، فإن استخدام جهاز كمومي حقيقي يحتاج إلى سبب علمي أو تعليمي واضح.

Before adding a new tool, establish a baseline. If a simple scikit-learn model performs stably, moving to a deep network should be justified by a measurable benefit. If a SQLite query takes fractions of a second, adding Spark is not automatically an improvement. If a classical simulation solves the example quickly and accurately enough, using real quantum hardware needs a clear scientific or educational reason.

1. زمن التنفيذ والذاكرة المستخدمة.

1. Execution time and memory use.

2. دقة النموذج أو الخطأ العددي.

2. Model accuracy or numerical error.

3. قابلية إعادة الإنتاج عبر تشغيلات مستقلة.

3. Reproducibility across independent runs.

4. وضوح المصدر والإصدار والمدخلات.

4. Clarity of data source, version, and inputs.

5. تكلفة التعقيد الإضافي مقارنة بالفائدة.

5. Cost of added complexity compared with benefit.

صندوق الكود — PYTHON

Code Box — PYTHON

```
from time import perf_counter

start = perf_counter()
result = sum(i * i for i in range(100_000))
elapsed = perf_counter() - start
print({"result": result, "seconds": elapsed})
```

Benchmark بسيط يصبح أكثر قيمة عندما تحفظ نتيجته مع نسخة البرنامج والبيانات. بهذه الطريقة يمكنك بعد شهر أن تعرف إن كان التغيير في السرعة سببه خوارزمية أفضل أم جهاز مختلف أم بيانات مختلفة.

Even a simple benchmark becomes more valuable when its result is stored with the software and dataset version. Months later, you can then determine whether a performance change came from a better algorithm, different hardware, or different data.

كتابة النتيجة العلمية: كيف تخاطب القارئ دون نبرة ادعاء؟ 14.22

14.22 Scientific Communication: How to Address the Reader Without Overclaiming

في مشروع متعدد التقنيات، قد يغريك اتساع الأدوات أن تصف النتيجة بعبارات أكبر من الدليل. الأفضل أن تفرق بين ثلاثة أفعال: «حسبنا»، و«اقترحنا»، و«يمكن أن نوسع». حسبنا عندما يوجد تنفيذ ونتيجة موثقة. اقترحنا عندما خرج نموذج بتنبؤ أو مرشح يحتاج إلى تحقق. ويمكن أن نوسع عندما نتحدث عن مسار مستقبلي لم ينفذ بعد.

In a multi-technology project, the breadth of tools can tempt you to describe results more strongly than the evidence allows. It is better to distinguish three actions: “we computed,” “we proposed,” and “we can extend.” Use “computed” when there is an executed, documented result. Use “proposed” for model predictions or candidates that still require validation. Use “can extend” for future work that has not yet been executed.

هذه اللغة ليست تقليلًا من قيمة المشروع؛ إنها تزيد قابليته للتصديق. عندما ترى نتيجة VQE، يجب أن تعرف هل هي محاكاة مثالية أم تشغيل على جهاز حقيقي. وعندما ترى اقتراحًا من نموذج توليدي، يجب أن تعرف هل هو مرشح عددي أم مركب ثبتت خواصه. وعندما ترى جسرًا إلى الجينوم، يجب أن تعرف أنه نموذج هندسي للبيانات وليس تحليلًا سريريًا.

This language does not diminish the project; it makes it more credible. When you see a VQE result, you should know whether it came from an ideal simulation or real hardware. When you see a generative-model proposal, you should know whether it is a numerical candidate or a compound with validated properties. When you see a bridge to genomics, you should know that it is a data-engineering model rather than a clinical analysis.

والقاعدة التي تجمع هذه الطبقات هي الأمانة في الحدود: لا تجعل صغر المثال يبدو كأنه جينوم كامل، ولا تجعل نموذجًا صغيرًا يبدو كأنه ذكاء اصطناعي عام، ولا تجعل محاكاة كمومية تعليمية تبدو كأنها تفوق عملي مثبت. القيمة هنا أن تتعلم كيف تبني المسار الذي يسمح لك بإثبات ذلك عندما تتوافر البيانات والأدوات المناسبة.

The rule connecting these layers is honesty about boundaries: do not make a small example look like a complete genome, do not make a small model look like general artificial intelligence, and do not make an educational quantum simulation look like a demonstrated practical advantage. The value lies in learning how to build the workflow that can establish those claims when appropriate data and tools become available.

14.23 اختبار الاتساق بين الطبقات

14.23 Cross-Layer Consistency Testing

عندما تتصل SQL بالمصفوفات والتعلم الآلي والكم، تظهر مشكلة جديدة: هل تمثل كل طبقة السجل نفسه؟ قد يكون العدد الذري في قاعدة البيانات صحيحًا، لكن ترتيب الصفوف في المصفوفة تغير، أو جرى تطبيق StandardScaler على بيانات التدريب والاختبار بطريقة مختلفة، أو استُخدمت نسخة أخرى من الميزات في مرحلة الكم. لذلك نحتاج اختبارات اتساق تعبر الحدود بين الطبقات.

When SQL connects to matrices, machine learning, and quantum workflows, a new problem appears: does every layer still represent the same record? The atomic number may be correct in the database while matrix row order changes, or StandardScaler may be applied inconsistently, or a different feature version may reach the quantum stage. Cross-layer consistency tests therefore become necessary.

صندوق الكود — PYTHON

Code Box — PYTHON

```
def assert_feature_contract(metadata, X):
    assert X.ndim == 2
    assert X.shape[0] == len(metadata)
    atomic_numbers = [m["atomic_number"] for m in metadata]
    assert atomic_numbers == sorted(atomic_numbers)
    assert len(set(atomic_numbers)) == len(atomic_numbers)

metadata = [{"atomic_number": 1}, {"atomic_number": 2}]
X = [[0.1, 1.0], [0.2, 1.1]]
import numpy as np
assert_feature_contract(metadata, np.asarray(X))
```

هذا النوع من الاختبار بسيط لكنه يوضح مبدأ مهمًا: لا يكفي أن تعمل كل دالة منفردة؛ يجب أن تعمل العقود بين الدوال والطبقات أيضًا. في مشروع أكبر، يمكن إضافة عملية الاختبار ليشمل إصدار الميزات، ترتيب الأعمدة، الوحدات، ومعرّف التجربة.

This kind of test is simple, but it illustrates an important principle: it is not enough for each function to work independently; the contracts between functions and layers must work as well. In a larger project, tests can cover feature versions, column order, units, and experiment identifiers.

14.24 سجل النَّسَب العلمي Provenance: من أين جاءت كل قيمة؟

14.24 Scientific Provenance: Where Did Every Value Come From?

في مشروع يجمع مصادر متعددة، لا ينبغي أن يكون المصدر حقلًا اختياريًا. لكل قيمة أو نتيجة مهمة يجب أن تستطيع الوصول إلى أصلها: مصدر البيانات، تاريخ الاستخراج، إصدار المصدر، التحويلات، والبرنامج الذي عالجها. هذا هو الفرق بين ملف نتائج قابل للعرض وسجل علمي قابل للمراجعة.

In a project that combines multiple sources, provenance should not be an optional field. For every important value or result, you should be able to trace its origin: data source, extraction date, source version, transformations, and software that processed it. That is the difference between a results file that can be displayed and a scientific record that can be audited.

صندوق الكود — SQL

Code Box — SQL

```
CREATE TABLE IF NOT EXISTS provenance (
    provenance_id INTEGER PRIMARY KEY,
    object_type TEXT NOT NULL,
    object_id TEXT NOT NULL,
    source_name TEXT NOT NULL,
    source_version TEXT NOT NULL,
    transformation TEXT NOT NULL,
    software_version TEXT NOT NULL,
    created_at TEXT NOT NULL
);

CREATE INDEX IF NOT EXISTS idx_provenance_object
ON provenance(object_type, object_id);
```

إذا وصلت نتيجة إلى التقرير النهائي دون سجل نسب، فقد تبدو صحيحة لكن يصعب التحقق منها. وعندما تكون النتيجة مرتبطة بقرار حساس، تصبح هذه الفجوة مشكلة عملية. لذلك ترتبط قابلية التتبع هنا مباشرة بالحوكمة والموثوقية، وليست مجرد إضافة توثيقية في نهاية المشروع.

If a result reaches the final report without provenance, it may look correct while remaining difficult to verify. When a result can influence a sensitive decision, that gap becomes practical. Provenance therefore connects directly to governance and reliability rather than being mere documentation added at the end.

مثال متعدد العناصر: لماذا لا نتوقف عند الهيدروجين؟ 14.25

14.25 A Multi-Element Example: Why Stop at Hydrogen?

الهيدروجين يشرح الحلقة الكمومية بأقل تعقيد، لكنه لا يمثل كل الأسئلة. يمكن استخدام الهيليوم مثالًا على ذرة لها إلكترونان، والليثيوم مثالًا على زيادة عدد الإلكترونات، والكربون والأكسجين كعناصر تدخل في جزيئات أكثر تعقيدًا. المقصود ليس أن نحسب كل هذه الأنظمة داخل الكتاب، بل أن ترى كيف يتغير حجم النموذج مع السؤال الفيزيائي.

Hydrogen explains the quantum loop with minimal complexity, but it does not represent every question. Helium can illustrate a two-electron atom, lithium illustrates an increase in electron count, and carbon and oxygen appear in more complex molecules. The purpose is not to compute all of these systems in the book, but to show how model size changes with the physical question.

في قاعدة البيانات يمكن أن تبقى العناصر الـ118 سجلات كاملة، بينما يختار المسار الكمومي عنصرًا أو جزيئًا واحدًا في كل تجربة. بهذه الطريقة لا نخلط بين «قاعدة بيانات العناصر» و«حجم مسألة المحاكاة». وجود 118 سجلًا لا يعني وجود 118 تجربة كمومية، ووجود تجربة H_2 لا يعني أن بقية العناصر غير مهمة.

In the database, all 118 elements can remain complete records, while the quantum path selects one element or molecule per experiment. This prevents confusion between the size of the element database and the size of a simulation problem. Having 118 records does not mean 118 quantum experiments, and having an H_2 experiment does not make the other elements irrelevant.

صندوق الكود — PYTHON

Code Box — PYTHON

```
ELEMENTS = {
    1: ("H", "Hydrogen"),
    2: ("He", "Helium"),
    3: ("Li", "Lithium"),
    6: ("C", "Carbon"),
    8: ("O", "Oxygen"),
}

for atomic_number, (symbol, name) in ELEMENTS.items():
    print(atomic_number, symbol, name)
```

يمكنك بعد ذلك ربط كل تجربة بمعرّف العنصر أو الجزيء، بدل تكرار البيانات داخل كل مرحلة. هذه ممارسة صغيرة لكنها تمثل فكرة OOP وSQL معًا: الهوية تُحفظ مرة، والعلاقات تُبنى حولها، والنتائج تشير إلى الأصل بدل نسخ الأصل في كل مكان.

You can then link each experiment to an element or molecule identifier instead of duplicating the data in every stage. This small practice expresses the OOP and SQL ideas together: identity is stored once, relationships are built around it, and results point back to the source instead of copying it everywhere.

14.26 ملف النتائج النهائي: ما الذي يجب أن يراه القارئ؟

14.26 The Final Results Record: What Should the Reader See?

التقرير النهائي الجيد لا يكتفي برقم نهائي. يجب أن يوضح المدخلات، النسخ البرمجية، المقياس المستخدم، حالة الاختبار، وما إذا كانت النتيجة محسوبة أو مقترحة أو غير منفذة. ويمكن أن يحمل التقرير تحذيرًا صريحًا عندما تكون طبقة الكم غير متاحة أو عندما تكون البيانات تجريبية تعليمية.

A good final report does not contain only a final number. It should show inputs, software versions, the metric used, test status, and whether a result was computed, proposed, or not executed. The report can also state explicitly when the quantum layer is unavailable or when the data are educational examples.

صندوق الكود — PYTHON

Code Box — PYTHON

```
report = {
    "dataset": "periodic-table-v1",
    "classical_pipeline": "executed",
    "ml_baseline": {"status": "executed", "metric": "documented"},
    "quantum_stage": {"status": "optional", "backend": "recorded"},
    "provenance": "required",
    "scientific_scope": "educational and engineering demonstration",
}

for key, value in report.items():
    print(f"{key}: {value}")
```

هذه البنية تجعل القارئ قادرًا على التمييز بين ما حدث بالفعل وما تم تصميمه كامتداد. وهي أيضًا مناسبة للأتمتة: يمكن لبرنامج التقرير أن يقرأ السجل ويولد تقريرًا لا يخفي فشل مرحلة أو غياب أداة.

This structure lets the reader distinguish what actually happened from what was designed as an extension. It also supports automation: a reporting program can read the record and produce a report that does not hide a failed stage or missing tool.

14.27 خلاصة التوسع العلمي قبل العودة إلى بقية المشروع

14.27 Scientific Expansion Summary Before Returning to the Rest of the Project

يمر عبر SQL، أصبح لديك الآن مسار أوسع من المثال الأول: يبدأ من 118 عنصرًا محفوظة في المصفوفات والجبر الخطي، ويستخدم التعلم الآلي والتوصية، ثم يوضح كيف يمكن أن تدخل النماذج التوليدية والبيانات الجينومية والحوسبة الموزعة، وأخيرًا ينتقل إلى أمثلة كمومية تتدرج من H_2 إلى H_2O وCO_2 مع هاملتوني وترميز وقيود موارد. هذا الاتساع لا يعني أن كل مرحلة نُفذت على جهاز حقيقي؛ بل يعني أن حدود كل مرحلة أصبحت واضحة وقابلة للتطوير.

You now have a broader workflow than the initial example: it starts with 118 elements stored in SQL, moves through matrices and linear algebra, uses machine learning and recommendation, then shows how generative models, genomic data, and distributed computing can enter, and finally moves to quantum examples ranging from H_2 to H_2O and CO_2 with Hamiltonians, mappings, and resource constraints. This breadth does not mean every stage has been executed on real hardware; it means the boundaries of each stage are explicit and extensible.

والأهم أن الجدول الدوري يظل في مكانه الصحيح: مثال تعليمي وهندسي مضبوط الحجم. أما الجينوم الحقيقي، فحجمه وتعقيده يجعلان تخزينه وتحليله واستدعاء ميزاته مسألة أكبر بكثير، وهو ما يبرر والحوسبة العلمية والذكاء الاصطناعي والحوسبة الكمومية عند توفر المسألة OOPو SQL الحاجة إلى المناسبة. لكن كل انتقال من طبقة إلى أخرى يحتاج إلى دليل، وإصدار، واختبار، وحدود واضحة للادعاء.

Most importantly, the periodic table remains in its proper role: a controlled-size educational and engineering example. A real genome is much larger and more complex, which helps explain why SQL, OOP, scientific computing, AI, and—when appropriate—quantum computing become relevant. Every transition between layers, however, still requires evidence, versioning, testing, and clear limits on the claim.

من المثال إلى المشروع القابل للإضافة عملية 14.28

14.28 From Example to an Extensible Project

إذا أردت تطوير المشروع بعد انتهاء الكتاب، فالأفضل ألا تبدأ بإضافة تقنيات جديدة بلا رابط. ابدأ بسؤال علمي محدد، ثم حدد البيانات المطلوبة، ثم اختر أبسط نموذج قادر على الإجابة، وبعد ذلك قرر هل تحتاج إلى تعلم عميق أو معالجة موزعة أو حوسبة كمومية. بهذه الطريقة يصبح كل توسع نتيجة لحاجة، لا مجرد إضافة اسم أداة إلى قائمة الأدوات.

If you want to extend the project after the book, do not begin by adding technologies without a link to a question. Start with a defined scientific question, identify the required data, choose the simplest model capable of answering it, and only then decide whether deep learning, distributed processing, or quantum computing is justified. Each extension should therefore answer a need rather than add another tool name to a list.

يمكنك أيضًا الاحتفاظ بنسخة صغيرة قابلة للتشغيل من المشروع إلى جانب البيانات الكبيرة. النسخة الصغيرة تؤدي دور اختبار الانحدار Regression Test: يجب أن إذا تغيرت المكتبات أو البنية، تعرف أولًا هل ما زالت النتائج الأساسية تعمل. وبعد نجاح الاختبار الصغير يمكن نقل التغيير إلى البيانات الأكبر. هذا الأسلوب يقلل مخاطر أن يؤدي تحديث بسيط إلى كسر خط أنابيب كامل.

You can also keep a small runnable version of the project beside any larger datasets. The small version serves as a regression test: when libraries or architecture change, you can first determine whether the core results still work. After the small test passes, the change can be moved to larger data. This reduces the risk that a small update breaks an entire pipeline.

صندوق الكود — PYTHON

Code Box — PYTHON

```
def regression_gate(current, expected, tolerance=1e-8):
    if len(current) != len(expected):
        return False
    return all(abs(a - b) <= tolerance for a, b in zip(current, expected))

baseline = [0.125, 0.250, 0.500]
new_run = [0.1250000001, 0.2499999999, 0.5000000000]
print("regression passed:", regression_gate(new_run, baseline))
```

هذا الاختبار لا يثبت صحة النموذج العلمي، لكنه يحميك من تغيير برمجي غير مقصود. ويجب أن يبقى الفرق بين اختبار صحة البرنامج وبين إثبات صحة الفرضية العلمية واضحًا. البرنامج يمكن أن يعمل كما صُمم، بينما تكون الفرضية أو البيانات نفسها بحاجة إلى مراجعة.

This test does not prove scientific validity, but it protects against unintended software changes. The distinction between software correctness and scientific validity should remain explicit. Software can work exactly as designed while the hypothesis or data still require review.

وهكذا يصبح الجزء الثالث منصة تعليمية وهندسية فعلية: ليس المطلوب أن تحفظ أسماء الأدوات، بل أن تعرف لماذا تدخل كل أداة، وما الذي تضيفه، وما الذي لا تستطيع إثباته، وكيف تحفظ نتيجة قابلة للتتبع. هذه هي الطريقة التي يمكن بها أن تنتقل من مثال صغير إلى مشروع أكبر دون أن تفقد البنية أو الأمانة العلمية.

The result is a genuinely educational and engineering platform: the goal is not to memorize tool names, but to understand why each tool enters, what it adds, what it cannot prove, and how to preserve a traceable result. This is how you can move from a small example to a larger project without losing structure or scientific integrity.

14.29 مثال متكامل قصير: من سجل العنصر إلى تجربة H_2O

14.29 A Compact End-to-End Example: From an Element Record to an H_2O Experiment

لنضع الطبقات كلها في مثال واحد دون الادعاء بأن الكتاب أجرى محاكاة كاملة للماء. يبدأ المسار من سجلات الهيدروجين والأكسجين في قاعدة العناصر، ثم يُنشأ سجل جزيئي يحدد أن التجربة تخص H_2O، ثم تُحفظ هندسة الجزيء والنموذج الإلكتروني والإصدار. بعد ذلك يمكن لطبقة الكيمياء الكمومية أن تنتج هاملتونيًا، ويمكن لطبقة التمثيل أن تحوله إلى مؤثرات كيوبتية، ويمكن لخوارزمية VQE أن تقدّر كمية محددة. النتيجة النهائية تعود إلى SQL مع سجل مصدرها.

Let us place all layers into one example without claiming that the book performed a complete water simulation. The path starts from hydrogen and oxygen records in the element database, then creates a molecular record identifying H_2O, stores its geometry, electronic model, and software version. A quantum-chemistry layer can then produce a Hamiltonian, a mapping layer can transform it into qubit operators, and VQE can estimate a defined quantity. The final result returns to SQL with its provenance record.

صندوق الكود — SQL

Code Box — SQL

```
INSERT INTO molecular_experiments (
    molecule, geometry, basis_set, active_space, mapper,
    algorithm, software_version, result_energy, result_unit, status, created_at
) VALUES (
    'H2O',
    'documented geometry',
    'documented basis',
    'documented active space',
    'Jordan-Wigner',
    'VQE',
    'recorded-version',
    NULL,
    'hartree',
    'planned',
    datetime('now')
);
```

قبل الحساب مقصود. فهو يمنع قاعدة البيانات من إعطاء انطباع بأن «planned» وضع الحالة
مع «executed» النتيجة حُسبت لمجرد وجود سجل التجربة. بعد التنفيذ يمكن أن تتحول الحالة إلى
مع سبب الفشل بدل «failed» رقم النتيجة ومعلومات الجهاز أو المحاكي. وإذا فشل التنفيذ تبقى الحالة
إنشاء قيمة وهمية.

The “planned” status is intentional. It prevents the database from implying that a result was computed merely because an experiment record exists. After execution, the status can become “executed” with the result and backend information. If execution fails, the status remains “failed” with a reason rather than creating a fabricated value.

هذا النمط الصغير يلخص فكرة المشروع كلها: البيانات لها هوية، والعمليات لها إصدارات، والنتائج لها سياق، والفشل نفسه معلومة. وعندما تكبر البيانات الجينومية أو تتصل المنصة بخدمة كمومية أو نموذج توليدي، لا تحتاج إلى اختراع فلسفة جديدة؛ بل تحتاج إلى إضافة عملية نفس العقود والهويات والتتبّع والاختبارات.

This small pattern summarizes the project: data have identity, operations have versions, results have context, and failure is itself information. When genomic data grow or the platform connects to a quantum service or generative model, you do not need a new philosophy; you extend the same contracts, identities, provenance, and tests.

14.30 آخر فحص قبل المتابعة: هل أضفت تقنية أم أضفت قدرة؟

14.30 Final Check Before Moving On: Did You Add a Technology or a Capability?

إذا أضفت Qiskit Runtime، فاسأل: هل أصبحت قادرًا على تسجيل تشغيل عتاد حقيقي بصورة أفضل؟ وإذا أضفت Dask، فاسأل: هل أصبحت قادرًا على معالجة بيانات أكبر من الذاكرة؟ وإذا أضفت VAE أو GAN، فاسأل: هل أصبحت قادرًا على استكشاف فضاء مرشحين مع تقييم واضح؟ وإذا أضفت Gene Resistance Tracker، فاسأل: هل أصبحت العلاقة بين الجين والمتغير والدليل قابلة للتتبع؟ السؤال الصحيح دائمًا هو القدرة الجديدة، لا اسم الأداة.

If you add Qiskit Runtime, ask: can you now record real-hardware execution more reliably? If you add Dask, ask: can you process data larger than memory? If you add a VAE or GAN, ask: can you explore a candidate space with explicit evaluation? If you add Gene Resistance Tracker, ask: can the relationship among gene, variant, and evidence be traced? The right question is always the new capability, not the tool name.

بهذا المعيار يمكن أن تبقى البنية بسيطة رغم اتساعها. ليست كل تقنية مرحلة إلزامية، وليست كل نتيجة تحتاج إلى جهاز متقدم. البنية القوية هي التي تعرف متى تستخدم الأداة، ومتى لا تستخدمها، وكيف تثبت أن القرار كان مناسبًا للمشكلة.

With this criterion, the architecture can remain simple even as it grows. Not every technology is mandatory, and not every result requires advanced hardware. A strong architecture knows when to use a tool, when not to use it, and how to demonstrate that the choice was appropriate for the problem.

الفصل الخامس عشر: التغليف التشغيلي وإعادة الإنتاج

Chapter 15: Operational Packaging and Reproducibility

15.1 من الكتاب إلى مشروع قابل لإعادة التشغيل

15.1 From a Book to a Reproducible Project

المرحلة الأخيرة من الهندسة التعليمية ليست إضافة مكتبة جديدة، بل تحويل المعرفة الموزعة في الفصول إلى بنية يستطيع قارئ آخر تشغيلها دون تخمين. لذلك يجب أن تكون لكل خطوة مدخلات محددة ومخرجات محددة وسجل واضح للإصدار. وجود ملف نصي يشرح الأمر لا يكفي إذا كان القارئ مطالبًا بإعادة كتابة الكود يدويًا. التغليف الجيد يفصل بين الشرح التعليمي وبين الأثر التنفيذي، لكنه لا يفصل بينهما من حيث التوثيق: الكود الموجود في الملفات يجب أن يكون هو النسخة التي يشرحها الكتاب.

The final stage of educational engineering is not adding another library; it is turning knowledge distributed across chapters into a structure another reader can execute without guesswork. Every stage should have explicit inputs, outputs, and version information. A text block that merely describes a command is not enough if the reader must retype the program manually. Good packaging separates instructional prose from executable artifacts without

separating them in provenance: the code shipped with the project should be the code explained by the book.

بنية المجلدات وحدود المسؤولية 15.2

15.2 Folder Structure and Separation of Responsibilities

البنية المقترحة تجعل قاعدة البيانات والبرامج والاختبارات والتقارير أجزاء مستقلة. يوضع مخطط SQL في scripts، وتوضع اختبارات pytest في tests، ويظل run_pipeline.py نقطة دخول واحدة. أما المخرجات فتذهب إلى data وartifacts وreports بدل الكتابة فوق المصدر. هذا الفصل بين المصدر والمخرجات يمنع خطأ شائعًا: تعديل ملف ناتج ثم اعتباره مصدرًا للحقيقة.

The proposed structure separates the database, scripts, tests, and reports. SQL schema files belong in scripts, pytest tests belong in tests, and run_pipeline.py provides one entry point. Generated outputs go to data, artifacts, and reports rather than overwriting source files. This separation prevents a common failure mode: editing a generated artifact and later treating it as the authoritative source.

التثبيت المقيد لا يعني التثبيت الأبدي 15.3

15.3 Pinned Environments Do Not Mean Permanent Environments

تثبيت الإصدارات يهدف إلى إعادة إنتاج تجربة محددة، وليس إلى الادعاء بأن تلك الإصدارات ستظل الأفضل إلى الأبد. في هذه النسخة التعليمية نحتفظ ببيئة مقيدة يمكن اختبارها، ونوثق تاريخها، ثم نميز بين بيئة إعادة الإنتاج وبيئة الترقية. هذا التفريق مهم خصوصًا في منظومة Qiskit Nature؛ فالإصدار 0.8.0 يرفع الحد الأدنى لـQiskit إلى 1.4 ويحتاج qiskit-algorithms بإصدار 0.4 أو أحدث، ولذلك يجب ألا تخلط أمثلة قديمة بواجهة حديثة دون مراجعة.

Pinning versions is intended to reproduce a defined experiment, not to claim that those versions will remain optimal forever. This educational edition therefore keeps a tested pinned environment, documents its date, and distinguishes reproducibility from future upgrade work. This distinction is especially important for Qiskit Nature: release 0.8.0 raises the minimum Qiskit version to 1.4 and

requires qiskit-algorithms 0.4 or newer, so legacy examples should not be mixed with a modern API without review.

الفصل السادس عشر: سلامة البيانات والتتبّع العلمي

Chapter 16: Data Integrity and Scientific Provenance

16.1 الهوية العلمية للسجل

16.1 Scientific Identity of a Record

كل عنصر في قاعدة البيانات يجب أن يملك هوية مستقرة: العدد الذري هو المفتاح الطبيعي الأقوى في هذا المثال، والرمز اختصار قابل للتحقق، والاسم والوزن الذري حقول وصفية تحتاج مصدرًا. عندما تنتقل البيانات إلى المصفوفة، لا ينبغي فقدان هذه الهوية؛ يجب أن يحتفظ التحليل بجدول metadata يربط كل صف بالعدد الذري. وإلا فقد يصبح رقم في المصفوفة صحيحًا حسابيًا لكنه منسوب إلى العنصر الخطأ.

Every element record needs a stable identity. Atomic number is the strongest natural key in this example, the symbol is a checkable abbreviation, and name and atomic weight are descriptive fields that need provenance. When data move into a matrix, that identity must not be lost; the analysis should retain metadata mapping every row back to atomic number. Otherwise a matrix value can be numerically correct yet assigned to the wrong element.

16.2 القيم المفقودة ليست مجرد فراغات

16.2 Missing Values Are Not Merely Empty Cells

معالجة NaN يجب أن تبدأ بتحديد سبب الغياب. قد تكون الخاصية غير مقاسة، أو غير منطبقة، أو غير متاحة في المصدر، أو فُقدت أثناء التحويل. التعويض العددي يخلق قيمة جديدة، ولذلك يجب تسجيل طريقة التعويض والنسخة الناتجة. في التحليل التعليمي يمكن استخدام المتوسط أو الوسيط أو طريقة robust، لكن لا يجوز تفسير القيمة المعوضة كقياس تجريبي أصلي.

Handling NaN should begin by identifying why the value is missing. It may be unmeasured, not applicable, unavailable in the source, or lost during transformation. Imputation creates a new value and therefore the method and resulting version should be recorded. Educational analysis may use mean, median, or robust approaches, but an imputed value must never be interpreted as an original experimental measurement.

16.3 مصدر البيانات جزء من البيانات

16.3 Provenance Is Part of the Data

السجل القابل للتدقيق لا يكتفي بالنتيجة. يحتاج إلى مصدر البيانات، وتاريخ الاستخراج، وإصدار البرنامج، والإعدادات التي أنتجت المصفوفة أو النموذج أو المحاكاة. في المشروع يمكن تمثيل ذلك بجدول provenance بسيط يربط artifact_id بالمصدر والنسخة ووقت التنفيذ. هذه الطبقة هي ما يجعل إعادة الإنتاج قابلة للفحص بدل أن تكون وعدًا عامًا.

An auditable record stores more than the result. It needs data source, extraction date, software version, and the settings that produced a matrix, model, or simulation. The project can represent this with a simple provenance table linking artifact_id to source, version, and execution time. This layer turns reproducibility from a general promise into something inspectable.

الفصل السابع عشر: التحقق العددي وحدود الاستنتاج

Chapter 17: Numerical Validation and Limits of Inference

17.1 التحقق ليس رقمًا واحدًا

17.1 Validation Is Not a Single Number

في التعلم الآلي لا يكفي عرض Accuracy أو R^2. يجب أن نعرف كيف قُسمت البيانات، وهل حدث تسرب بين التدريب والاختبار، وهل تتغير النتيجة بتغير البذرة العشوائية، وهل المقياس مناسب لطبيعة الهدف. ومع مجموعة من 118 عنصرًا فقط، ينبغي أن يكون الادعاء متواضعًا: النماذج هنا أدوات استكشاف وتعليم، وليست برهانًا على قانون كيميائي عام.

Machine-learning validation is not satisfied by reporting Accuracy or R^2 alone. We need to know how data were split, whether leakage occurred, whether results change with the random seed, and whether the metric matches the target. With only 118 elements, claims should remain modest: these models are exploratory and educational tools, not proofs of universal chemical laws.

17.2 اختبار الاستقرار عبر البذور

17.2 Stability Across Random Seeds

إذا كان النموذج يعتمد على تقسيم عشوائي أو خوارزمية عشوائية، فمن المفيد تكرار التجربة عبر عدة seeds ومقارنة المتوسط والانحراف المعياري. هذه الخطوة لا تجعل النموذج صحيحًا تلقائيًا، لكنها تكشف ما إذا كانت النتيجة حساسة لتقسيم واحد. إذا كان الاختلاف كبيرًا، فهذه إشارة إلى أن البيانات قليلة أو أن النموذج غير مستقر بالنسبة للمهمة المحددة.

When a model uses random splitting or a stochastic algorithm, repeating the experiment across several seeds and reporting the mean and standard deviation is useful. This does not make the model automatically correct, but it reveals whether the result

depends heavily on one split. Large variation suggests that the dataset is small or that the model is unstable for the stated task.

الارتباط لا يساوي السببية 17.3

17.3 Correlation Is Not Causation

أو التجميع أو نموذج تعلم آلي نمطًا بين الوزن الذري والتصنيف أو خاصية أخرى. هذا النمط لا يعني أن PCA قد يجد أحد المتغيرين سبب للآخر. في المشروع يجب أن تظل اللغة دقيقة: نقول ارتباطًا أو قدرة تنبؤية داخل البيانات المستخدمة، ولا نقول تأثيرًا سببيًا إلا إذا كان التصميم التجريبي أو الأدلة المستقلة يسمحان بذلك.

PCA, clustering, or a machine-learning model may reveal a pattern between atomic weight, category, or another feature. That pattern does not establish causation. The project therefore uses precise language: association or predictive performance within the analyzed data, not causal effect unless an appropriate experimental design or independent evidence supports it.

الفصل الثامن عشر: المحاكاة الكمومية من المثال إلى التحقق

Chapter 18: From Quantum Demonstration to Verification

لماذا يبدأ المثال بـ H_2؟ 18.1

18.1 Why Start with H_2?

ليس جينومًا ولا نموذجًا كاملًا لكل الكيمياء. إنه مسألة صغيرة يمكن فيها شرح دورة العمل من بناء المشكلة H_2 الإلكترونية إلى الهاملتوني، ثم خريطة الكيوبت، والدائرة، والمحسن، وقياس الطاقة، وأخيرًا المقارنة مع مرجع كلاسيكي. صغر المثال هنا ميزة تعليمية لأنه يسمح للقارئ بفحص كل خطوة بدل إخفاء التفاصيل داخل نظام ضخم.

H_2 is not a genome and is not a complete model of chemistry. It is a small problem that lets the reader follow the workflow from an electronic-structure problem to a Hamiltonian, qubit mapping,

circuit, optimizer, energy estimate, and classical comparison. Its small size is an educational advantage because each step can be inspected rather than hidden inside a large system.

18.2 الانتقال إلى H_2O وCO_2

18.2 Extending to H_2O and CO_2

زيادة عدد الكيوبتات من 2 إلى 4 أو 6 ليست وحدها توسعًا علميًا. يجب تحديد هندسة الجزيء، والمسافات بين الذرات، وقاعدة الأساس، ونوع التقريب، وعدد الإلكترونات الفعالة، وطريقة mapping، ثم اختيار ansatz ومحسن مناسبين. بعد ذلك فقط يصبح من المنطقي مقارنة VQE بمرجع كلاسيكي مستقل. لذلك يعرض المشروع H_2O وCO_2 كامتداد قابل للبناء والاختبار، لا كنتائج مكتملة لم تُحسب داخل النسخة الحالية.

Increasing the number of qubits from 2 to 4 or 6 is not, by itself, a scientific extension. Molecular geometry, bond lengths, basis set, approximation, active-electron space, mapping, ansatz, and optimizer must be specified first. Only then does comparison with an independent classical reference become meaningful. The project therefore treats H_2O and CO_2 as buildable and testable extensions, not as completed results that were never actually computed in this edition.

18.3 HHL وحدود صياغته

18.3 HHL and the Limits of Its Formulation

خوارزمية HHL تحل أنظمة خطية كمومية تحت شروط رياضية محددة. لا يصح تقديمها للقارئ كأداة مباشرة لحل معادلة شرودنجر في كل حالة. يلزم تحويل المسألة إلى نظام خطي مناسب، وتمثيل المصفوفة والحالة، وتحضير الحالة، ثم استخراج كمية قابلة للقياس. لذلك يرد HHL هنا كمسار بحثي متقدم يوضح الفرق بين اسم خوارزمية وبين صلاحية استخدامها لمسألة فيزيائية بعينها.

HHL solves quantum linear systems under specific mathematical assumptions. It should not be presented as a universal button for solving the Schrödinger equation. The problem must first be formulated as a suitable linear system, with matrix representation,

state preparation, and an observable quantity to extract. HHL is therefore presented as an advanced research direction that illustrates the difference between naming an algorithm and establishing its suitability for a particular physical problem.

الفصل التاسع عشر: الذكاء الاصطناعي التوليدي والاقتراحات القابلة للمراجعة

Chapter 19: Generative AI and Reviewable Proposals

19.1 من التنبؤ إلى التوليد

19.1 From Prediction to Generation

النموذج التنبؤي يتعلم علاقة بين مدخلات ومخرج معروف نسبيًا، بينما النموذج التوليدي يقترح مخرجات جديدة ضمن فضاء تعلمه. لذلك تختلف المخاطر المعرفية: قد ينتج النموذج اقتراحًا يبدو معقولًا لغويًا أو عدديًا لكنه غير مستقر كيميائيًا أو غير قابل للتصنيع. يجب أن يمر الاقتراح عبر فحوص مستقلة قبل اعتباره نتيجة علمية.

A predictive model learns a relationship between inputs and a relatively defined target, while a generative model proposes new outputs within a learned space. The epistemic risks therefore differ: a generated proposal can look plausible while being chemically unstable or experimentally impractical. Every proposal should pass independent checks before being treated as scientific evidence.

19.2 GAN وVAE كنماذج اقتراحية

19.2 GANs and VAEs as Proposal Models

يمكن استخدام GANs وVAEs لتعلم تمثيل مضغوط للخصائص أو لتوليد عينات مرشحة. في هذا المشروع يجب أن تكون البيانات التدريبية محددة، وأن تُفصل بيانات التقييم عن التدريب، وأن يسجل النظام سبب اختيار كل خاصية. كما يجب أن يظل الناتج موسومًا بوضوح على أنه generated candidate؛ هذه التسمية ليست شكلية؛ لا مادة مثبتة. إنها جزء من سلسلة الأدلة.

GANs and VAEs can learn compact representations of properties or generate candidate samples. In this project, training data must be defined, evaluation data kept separate, and the reason for each selected feature documented. Outputs must remain explicitly labeled as generated candidates rather than validated materials. That label is not cosmetic; it is part of the evidence chain.

19.3 نظام التوصية متعدد القيود

19.3 A Multi-Constraint Recommendation System

نظام التوصية لا ينبغي أن يعيد عنصرًا واحدًا بناءً على خاصية واحدة. يمكن أن يحسب درجة مركبة من أهداف محددة، ثم يعرض أفضل المرشحين مع القيود التي سببت الترتيب. إذا تعارضت الأهداف، يجب إظهار التعارض بدل إخفائه داخل رقم واحد. بهذه الطريقة يصبح النظام أداة قرار قابلة للمراجعة، لا صندوقًا أسود يعطي توصية نهائية.

A recommendation system should not return one element based on a single property. It can compute a composite score over explicit objectives and show the top candidates together with the constraints that produced the ranking. When objectives conflict, the conflict should be visible rather than hidden inside one number. This turns the recommender into a reviewable decision-support tool rather than a black box issuing a final answer.

الفصل العشرون: الجسر إلى البيانات الجينومية دون ادعاء أن الجدول الدوري جينوم

Chapter 20: Bridging to Genomic Data Without Pretending the Periodic Table Is a Genome

لماذا نحتاج الجسر؟ 20.1

20.1 Why Build the Bridge?

والمصفوفات والتعلم الآلي SQL الجزء الثالث يستخدم الجدول الدوري لأنه مجموعة صغيرة ومنظمة تصلح لتعليم والحوسبة الكمومية في كتاب واحد. أما الجينوم الحقيقي فيضم تسلسلات وقراءات وتعليقات وميزات متعددة، ويمكن أن يصبح تدفق البيانات ضخمًا. لذلك يستخدم المشروع الجدول الدوري كنموذج هندسي صغير، ثم يوضح كيف يمكن أن تمتد البنية إلى بيانات البروتينات والإنزيمات والتسلسلات دون الادعاء بأن المثال الحالي يمثل جينومًا كاملًا.

Part III uses the periodic table because it is a small, structured dataset that can teach SQL, matrices, machine learning, and quantum computing within one volume. A real genome involves reads, assemblies, annotations, and many feature layers and can become very large. The project therefore uses the periodic table as a compact engineering model and then explains how the architecture could extend to proteins, enzymes, and sequences without claiming that the current example represents a complete genome.

20.2 Gene Resistance Tracker بوصفه مخطط بيانات

20.2 Gene Resistance Tracker as a Data Schema

الاسم هنا يصف نموذجًا مفاهيميًا للبيانات، وليس أداة سريرية أو بروتوكولًا للتعديل الجيني. يمكن أن يحتوي النموذج على sample_id وsequence_id وvariant_id وevidence_id وanalysis_id، ثم يربطها بمصادر ونسخ ونتائج. القيمة التعليمية هي إظهار كيف يتحول سؤال بيولوجي إلى بنية قابلة للتدقيق، مع الحفاظ على الفصل بين البيانات الخام والاستنتاجات.

The name here describes a conceptual data model, not a clinical tool or a gene-editing protocol. A schema could include sample_id, sequence_id, variant_id, evidence_id, and analysis_id, linked to sources, versions, and results. The educational value is showing how a biological question can become an auditable data structure while keeping raw data separate from interpretation.

الخصوصية وحماية البيانات الحيوية 20.3

20.3 Privacy and Protection of Biological Data

عندما تنتقل البنية من الجدول الدوري إلى عينات بشرية، تتغير مسؤولية النظام. قد تصبح البيانات قابلة للربط بأفراد أو أسر، ولذلك يجب تقليل البيانات، وضبط الصلاحيات، وتسجيل الوصول، وفصل الهوية عن التحليل، وتحديد سياسة الاحتفاظ. هذه ليست إضافة فلسفية؛ إنها متطلبات هندسية إذا كان النظام سيعمل على بيانات بشرية حقيقية.

When the architecture moves from the periodic table to human samples, the system's responsibilities change. Data may become linkable to individuals or families, requiring data minimization, access control, access logging, identity separation, and retention policies. These are not philosophical extras; they are engineering requirements for systems handling real human biological data.

الفصل الحادي والعشرون: خط أنابيب متكامل قابل للتدقيق

Chapter 21: An Integrated Auditable Pipeline

مراحل الخط المتكامل 21.1

21.1 Stages of the Integrated Pipeline

يمكن تلخيص المشروع في خط واحد: تحميل البيانات، التحقق، بناء المصفوفة، إنشاء الميزات، التدريب، التقييم، التمثيل الكمومي، التحقق الكلاسيكي، التخزين، ثم التقرير. لكن هذا الخط لا يعني تنفيذ كل شيء في عملية واحدة غير قابلة للتجزئة. كل مرحلة يجب أن تكون مستقلة وقابلة لإعادة التشغيل، ويجب أن تعرف متى تستخدم cache ومتى تعيد الحساب.

The project can be summarized as one pipeline: load data, validate, build the matrix, create features, train, evaluate, encode quantum states, verify classically, persist results, and report. This does not mean one monolithic operation. Each stage should be independently rerunnable, with explicit rules for caching and recomputation.

سجل الأحداث والأخطاء 21.2

21.2 Event and Error Logging

التسجيل الجيد لا يطبع رسائل كثيرة فقط. يجب أن يحدد وقت البداية والنهاية، المرحلة، إصدار الكود، المدخلات الأساسية، مستوى الخطأ، ومكان المخرجات. إذا فشلت مرحلة الكم، يجب أن يذكر السجل السبب ويظل بإمكان المستخدم تشغيل المراحل الكلاسيكية. وإذا نجحت المرحلة، يجب أن تسجل checksum أو معرفًا للمخرج عندما يكون ذلك مناسبًا.

Good logging is not simply printing many messages. It records start and end times, stage, code version, key inputs, error level, and output location. If the quantum stage fails, the log should state why while allowing classical stages to run. When a stage succeeds, an output checksum or identifier can be recorded where appropriate.

التقرير كمنتج قابل للمراجعة 21.3

21.3 The Report as a Reviewable Artifact

التقرير النهائي يجب أن يذكر البيانات المستخدمة، والإصدارات، والمقاييس، ونتيجة الاختبارات، وأي تحذيرات. لا ينبغي أن يخفي فشلًا أو نتيجة غير مستقرة. التقرير الجيد يسمح للقارئ أن يفرق بين ما تم حسابه فعلًا وما هو مثال توضيحي أو امتداد مستقبلي. بهذه الطريقة يصبح الكتاب والمشروع متطابقين في حدود الادعاء العلمي.

The final report should state the data used, software versions, metrics, test results, and warnings. It should not hide failures or unstable outcomes. A good report lets the reader distinguish what was actually computed from what is illustrative or future work. This

keeps the book and the executable project aligned in their scientific claims.

الملحق التشغيلي: الملفات التنفيذية الكاملة للمشروع

Operational Appendix: Complete Executable Project Files

ملف README.md

File: README.md

محتوى الملف التنفيذي — قابل للنسخ مباشرة دون إعادة صياغة.

Executable file content — designed for direct extraction without reformatting.

MARKDOWN

```markdown
# Brick by Brick — Part III

This project is an educational and engineering workflow built around the 118-element periodic table. It
demonstrates SQL data modeling, numerical analysis, machine learning, quantum-state representation,
VQE for H2, persistent quantum results, verification, testing, provenance, and governance.

## Installation

Use Python 3.12 for the pinned environment. Create a virtual environment, install requirements.txt, then
run the pipeline. The quantum stage can be skipped when Qiskit/PySCF are unavailable.

## Execution

python run_pipeline.py
python run_pipeline.py --skip-quantum
python run_pipeline.py --stage database
python run_pipeline.py --stage ml
python -m pytest -q

## Structure

- scripts/build_database.sql
- scripts/insert_elements.py
- scripts/train_models.py
- scripts/generate_report.py
- tests/test_periodic_table.py
- data/
- artifacts/
- reports/
```

ملف requirements.txt

File: requirements.txt

محتوى الملف التنفيذي — قابل للنسخ مباشرة دون إعادة صياغة.

Executable file content — designed for direct extraction without reformatting.

TXT

```
numpy==2.3.5
pandas==2.2.3
scikit-learn==1.8.0
qiskit==2.5.0
qiskit-algorithms==0.4.0
qiskit-nature==0.8.0
pyscf==2.14.0
pytest==8.4.1
dask==2026.7.1
```

ملف environment.yml

File: environment.yml

محتوى الملف التنفيذي — قابل للنسخ مباشرة دون إعادة صياغة.

Executable file content — designed for direct extraction without reformatting.

YAML

```
name: brick-by-brick-part-iii
channels:
  - conda-forge
dependencies:
  - python=3.12
  - pip
  - numpy=2.3.5
  - pandas=2.2.3
  - scikit-learn=1.8.0
  - pytest=8.4.1
  - pip:
      - qiskit==2.5.0
      - qiskit-algorithms==0.4.0
      - qiskit-nature==0.8.0
      - pyscf==2.14.0
      - dask==2026.7.1
```

ملف scripts/build_database.sql

File: scripts/build_database.sql

محتوى الملف التنفيذي — قابل للنسخ مباشرة دون إعادة صياغة.

Executable file content — designed for direct extraction without reformatting.

SQL

```sql
PRAGMA foreign_keys = ON;

CREATE TABLE IF NOT EXISTS groups (
  group_number INTEGER PRIMARY KEY,
  group_name TEXT NOT NULL,
  group_english TEXT NOT NULL
);

CREATE TABLE IF NOT EXISTS periods (
  period_number INTEGER PRIMARY KEY,
  period_name TEXT NOT NULL
);

CREATE TABLE IF NOT EXISTS categories (
  category_id INTEGER PRIMARY KEY,
  name TEXT NOT NULL UNIQUE,
  name_english TEXT NOT NULL
);

CREATE TABLE IF NOT EXISTS elements (
  atomic_number INTEGER PRIMARY KEY,
  symbol TEXT NOT NULL UNIQUE,
  name TEXT NOT NULL,
  name_english TEXT NOT NULL,
  atomic_weight REAL NOT NULL,
  group_number INTEGER,
  period_number INTEGER,
  category_id INTEGER,
  FOREIGN KEY(group_number) REFERENCES groups(group_number),
  FOREIGN KEY(period_number) REFERENCES periods(period_number),
  FOREIGN KEY(category_id) REFERENCES categories(category_id)
);

CREATE INDEX IF NOT EXISTS idx_elements_group ON elements(group_number);
CREATE INDEX IF NOT EXISTS idx_elements_period ON elements(period_number);
CREATE INDEX IF NOT EXISTS idx_elements_category ON elements(category_id);
```

ملف scripts/insert_elements.py

File: scripts/insert_elements.py

محتوى الملف التنفيذي — قابل للنسخ مباشرة دون إعادة صياغة.

Executable file content — designed for direct extraction without reformatting.

PYTHON

```
from pathlib import Path
import sqlite3

DB = Path("data/periodic_table.db")

def validate_rows(rows):
    assert len(rows) == 118, f"Expected 118 elements, got {len(rows)}"
    symbols = [r[1] for r in rows]
    assert len(symbols) == len(set(symbols)), "Duplicate element symbols detected"

def load(rows):
    DB.parent.mkdir(parents=True, exist_ok=True)
    with sqlite3.connect(DB) as con:
        con.execute("PRAGMA foreign_keys = ON")
        with con:
            con.executemany(
                "INSERT OR REPLACE INTO
elements(atomic_number,symbol,name,name_english,atomic_weight,group_number,period_number,categ
ory_id) VALUES (?,?,?,?,?,?,?,?)",
                rows,
            )

if __name__ == "__main__":
    raise SystemExit("Populate rows from the project data source before execution; the validator requires
exactly 118 records.")
```

ملف scripts/train_models.py

File: scripts/train_models.py

محتوى الملف التنفيذي — قابل للنسخ مباشرة دون إعادة صياغة.

Executable file content — designed for direct extraction without reformatting.

PYTHON

```
from pathlib import Path
import sqlite3
import json
import pandas as pd
from sklearn.ensemble import RandomForestRegressor, RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, r2_score, accuracy_score, confusion_matrix

DB = Path("data/periodic_table.db")
OUT = Path("artifacts/model_metrics.json")

def main():
    with sqlite3.connect(DB) as con:
        df = pd.read_sql_query("SELECT atomic_number, atomic_weight, group_number, period_number
FROM elements ORDER BY atomic_number", con)
    X = df[["atomic_number", "group_number", "period_number"]]
    y_reg = df["atomic_weight"]
    Xtr, Xte, ytr, yte = train_test_split(X, y_reg, test_size=0.2, random_state=42)
    reg = RandomForestRegressor(n_estimators=300, random_state=42)
    reg.fit(Xtr, ytr)
    pred = reg.predict(Xte)
```

```
    metrics = {"mae": float(mean_absolute_error(yte, pred)), "r2": float(r2_score(yte, pred))}
    # Classification is illustrative: classify by periodic-table group bands.
    y_cls = (df["group_number"] >= 13).astype(int)
    Xtr, Xte, ytr, yte = train_test_split(X, y_cls, test_size=0.2, random_state=42, stratify=y_cls)
    clf = RandomForestClassifier(n_estimators=300, random_state=42)
    clf.fit(Xtr, ytr)
    cp = clf.predict(Xte)
    metrics.update({"accuracy": float(accuracy_score(yte, cp)), "confusion_matrix": confusion_matrix(yte,
cp).tolist()})
    OUT.parent.mkdir(parents=True, exist_ok=True)
    OUT.write_text(json.dumps(metrics, indent=2), encoding="utf-8")
    print(json.dumps(metrics, indent=2))

if __name__ == "__main__":
    main()
```

ملف scripts/generate_report.py

File: scripts/generate_report.py

محتوى الملف التنفيذي — قابل للنسخ مباشرة دون إعادة صياغة.

Executable file content — designed for direct extraction without reformatting.

PYTHON

```
from pathlib import Path
import json
from datetime import datetime, timezone

METRICS = Path("artifacts/model_metrics.json")
REPORT = Path("reports/final_report.md")

def main():
    REPORT.parent.mkdir(parents=True, exist_ok=True)
    metrics = json.loads(METRICS.read_text(encoding="utf-8")) if METRICS.exists() else {}
    lines = [
        "# Part III Computational Report",
        "",
        f"Generated: {datetime.now(timezone.utc).isoformat()}",
        "",
        "## Model metrics",
        "",
        "```json",
        json.dumps(metrics, indent=2),
        "```",
        "",
        "## Reproducibility",
        "",
        "The report must be interpreted together with the pinned environment and test results.",
    ]
    REPORT.write_text("\n".join(lines), encoding="utf-8")

if __name__ == "__main__":
    main()
```

ملف tests/test_periodic_table.py

File: tests/test_periodic_table.py

محتوى الملف التنفيذي — قابل للنسخ مباشرة دون إعادة صياغة.

Executable file content — designed for direct extraction without reformatting.

PYTHON

```
from pathlib import Path
import sqlite3
import numpy as np

DB = Path("data/periodic_table.db")

def test_database_exists():
    assert DB.exists()

def test_elements_count():
    with sqlite3.connect(DB) as con:
        n = con.execute("SELECT COUNT(*) FROM elements").fetchone()[0]
    assert n == 118

def test_unique_symbols():
    with sqlite3.connect(DB) as con:
        n, u = con.execute("SELECT COUNT(symbol), COUNT(DISTINCT symbol) FROM
elements").fetchone()
    assert n == u

def test_matrix_shape():
    with sqlite3.connect(DB) as con:
        rows = con.execute("SELECT atomic_number, atomic_weight, group_number, period_number FROM
elements ORDER BY atomic_number").fetchall()
    X = np.asarray([r[1:] for r in rows], dtype=float)
    assert X.shape[0] == 118
    assert X.shape[1] == 3

def test_matrix_has_no_nan_after_imputation():
    with sqlite3.connect(DB) as con:
        rows = con.execute("SELECT atomic_weight, group_number, period_number FROM elements ORDER
BY atomic_number").fetchall()
    X = np.asarray(rows, dtype=float)
    assert not np.isnan(X).any()

def test_quantum_encoder_encodes_hydrogen():
    # The encoder contract: hydrogen is atomic number 1 and must produce a normalized vector.
    v = np.array([1.0, 1.008], dtype=float)
    v = v / np.linalg.norm(v)
    assert np.isclose(np.linalg.norm(v), 1.0)
```

ملف run_pipeline.py

File: run_pipeline.py

محتوى الملف التنفيذي — قابل للنسخ مباشرة دون إعادة صياغة.

Executable file content — designed for direct extraction without reformatting.

PYTHON

```python
from __future__ import annotations
import argparse
import logging
import subprocess
import sys
from pathlib import Path

logging.basicConfig(level=logging.INFO, format="%(asctime)s | %(levelname)s | %(message)s")

STAGES = {
    "database": [sys.executable, "scripts/insert_elements.py"],
    "ml": [sys.executable, "scripts/train_models.py"],
    "report": [sys.executable, "scripts/generate_report.py"],
}

def run_stage(name: str):
    logging.info("Starting stage: %s", name)
    subprocess.run(STAGES[name], check=True)
    logging.info("Completed stage: %s", name)

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument("--skip-quantum", action="store_true")
    parser.add_argument("--stage", choices=["database", "ml", "report", "all"], default="all")
    args = parser.parse_args()
    if args.stage == "all":
        for name in ("database", "ml", "report"):
            run_stage(name)
        if args.skip_quantum:
            logging.info("Quantum stage skipped by request")
        else:
            logging.info("Quantum stage is optional and must be enabled in a validated quantum
environment")
    else:
        run_stage(args.stage)

if __name__ == "__main__":
    main()
```

قائمة التحقق النهائية قبل النشر

Final Pre-Publication Checklist

تم الحفاظ على الرسومات التعليمية الموجودة، بما فيها لوحة الجدول الدوري الكاملة ولوحات OOP→SQL، الجبر الخطي، الذكاء الاصطناعي، الحلقة الكمومية الهجينة، التحقق الكمومي، والتتبّع.

The existing educational graphics are preserved, including the complete periodic-table plate and the OOP→SQL, linear-algebra, AI, hybrid quantum, quantum-verification, and provenance plates.

وبقية VQE، الجدول الدوري مثال تعليمي منظم وليس تمثيلًا لجينوم كامل. الهيدروجين هو المثال المستخدم في مسار العناصر سجلات تحليلية وليست جينومًا مصغرًا.

The periodic table is a compact educational dataset, not a complete genome. Hydrogen is the explicit VQE example, while the other elements remain analytical records rather than a miniature genome.

كل فقرة تعليمية جديدة أضيفت بالعربية ثم الإنجليزية مباشرة، مع إبقاء الكود في صناديق قابلة للاستخراج وبمحاذاة يسارية.

Every newly added instructional block follows Arabic then English directly, while code remains in extractable left-aligned code boxes.

تم فصل الاقتراح التوليدي عن الإثبات العلمي، وتم فصل التمثيل الكمومي للميزات عن الادعاء بأنه نموذج فيزيائي كامل.

Generative proposals are separated from scientific evidence, and quantum feature encoding is separated from claims of a complete physical model.

تم تثبيت بيئة قابلة لإعادة الإنتاج مع التنبيه إلى أن التثبيت لقطة زمنية وليس ضمانًا دائمًا للتوافق المستقبلي.

A reproducible pinned environment is provided, with the explicit caveat that a pin is a time-specific reproducibility snapshot, not a permanent compatibility guarantee.

الفصل الثاني والعشرون: هندسة SQL والأداء وقابلية التوسع

Chapter 22: SQL Engineering, Performance, and Scalability

22.1 الفهارس ليست زينة

22.1 Indexes Are Not Decoration

الفهرس بنية إضافية تسمح لمحرك قاعدة البيانات بتقليل مساحة البحث عندما يكون الاستعلام مناسبًا له. لكنه ليس تحسينًا مجانيًا: كل فهرس يحتاج مساحة ويضيف كلفة إلى عمليات INSERT وUPDATE وDELETE. في قاعدة صغيرة من

118 عنصرًا قد لا يظهر الفرق زمنيًا، لكن المثال يوضح مبدأً يصبح مهمًا عندما ينتقل التصميم إلى ملايين أو مليارات السجلات. لذلك يجب أن يُبنى الفهرس من نمط الاستعلام الفعلي، لا من الرغبة في جعل قاعدة البيانات تبدو أكثر تعقيدًا.

An index is an additional structure that can reduce the search space when a query can use it. It is not a free optimization: every index consumes storage and adds work to INSERT, UPDATE, and DELETE operations. On a database of only 118 elements the timing difference may be negligible, but the example teaches a principle that becomes important at millions or billions of records. Indexes should therefore be designed from actual query patterns rather than from a desire to make a database look more sophisticated.

22.2 خطة التنفيذ و EXPLAIN QUERY PLAN

22.2 Execution Plans with EXPLAIN QUERY PLAN

الخطوة الهندسية التالية هي سؤال المحرك عن الطريقة التي يعتزم بها تنفيذ الاستعلام. في SQLite يمكن استخدام EXPLAIN QUERY PLAN لمعرفة ما إذا كان الاستعلام يستخدم فهرسًا أو يمر على الجدول كاملًا. لا ينبغي تفسير كلمة SEARCH أو SCAN وحدها على أنها حكم مطلق؛ يجب قراءة الخطة مع حجم البيانات وانتقائية الشرط وكلفة الفهرس. هذه عادة مهمة لأنها تنقل تحسين الأداء من التخمين إلى الملاحظة.

The next engineering step is to ask the database engine how it intends to execute a query. SQLite provides EXPLAIN QUERY PLAN to reveal whether an index can be used or whether the table is scanned. A word such as SEARCH or SCAN is not by itself a universal verdict; the plan must be interpreted together with data volume, predicate selectivity, and index cost. This moves performance work from guesswork to observation.

22.3 المعاملات وحدود الاتساق

22.3 Transactions and Consistency Boundaries

عند إدخال 118 عنصرًا أو تحديث مجموعة مترابطة من السجلات، تمنح المعاملة قاعدة البيانات نقطة قرار واضحة: إما أن تكتمل مجموعة التغييرات، أو تعود إلى الحالة السابقة عند الفشل. هذا لا يضمن صحة المحتوى نفسه؛ إنه يضمن

اتساق عملية الكتابة. لذلك يجمع المشروع بين القيود والمعاملات والتحقق بعد الإدخال، لأن كل طبقة تحمي نوعًا مختلفًا من الأخطاء.

When loading 118 elements or updating a related group of records, a transaction gives the database a clear decision boundary: either the change set commits, or the database returns to the previous state on failure. A transaction does not guarantee that the content itself is scientifically correct; it guarantees the consistency of the write operation. The project therefore combines constraints, transactions, and post-load validation because each layer protects against a different class of error.

22.4 من قاعدة صغيرة إلى نظام كبير

22.4 From a Small Database to a Larger System

إذا انتقل التصميم إلى بيانات جينومية، ستظهر احتياجات جديدة: partitioning، bulk loading، materialized summaries، وربما محركات تخزين مختلفة. لكن الانتقال لا يبدأ بإلغاء SQL أو القفز إلى بنية موزعة؛ يبدأ بقياس الاختناق الحقيقي. قد تكون المشكلة في الاستعلام، أو الفهرس، أو التخزين، أو الشبكة، أو المعالجة اللاحقة. هذه طريقة تفكير قابلة للنقل من مثال الجدول الدوري إلى أنظمة علمية أكبر.

If the design grows into genomic data, new needs may appear: partitioning, bulk loading, materialized summaries, and possibly different storage engines. The transition should not begin by abandoning SQL or jumping immediately to a distributed architecture; it begins by measuring the real bottleneck. The problem may be the query, index, storage, network, or downstream computation. This reasoning transfers directly from the periodic-table example to larger scientific systems.

الفصل الثالث والعشرون: هندسة الميزات والتحقق في التعلم الآلي

Chapter 23: Feature Engineering and Machine-Learning Validation

الميزة الجيدة ليست مجرد رقم 23.1

23.1 A Good Feature Is More Than a Number

الميزة الجيدة تحمل إشارة مرتبطة بالسؤال وتملك تعريفًا واضحًا. العدد الذري، المجموعة، والدورة لها معنى بنيوي في الجدول الدوري، بينما ميزة مصطنعة بلا تفسير قد تجعل النموذج يعمل ظاهريًا لكنها تضعف قابلية الفهم. كل ميزة ينبغي أن تحمل اسمًا ونوعًا ومصدرًا ووحدة أو تعريفًا، وأن يكون واضحًا متى تم حسابها.

A useful feature carries a signal relevant to the question and has a clear definition. Atomic number, group, and period have structural meaning in the periodic table, whereas an arbitrary feature may improve a score while weakening interpretability. Every feature should have a name, type, source, unit or definition, and a documented computation time.

منع تسرب البيانات 23.2

23.2 Preventing Data Leakage

تسرب البيانات يحدث عندما تدخل معلومة من مجموعة الاختبار إلى عملية تدريب النموذج بصورة مباشرة أو غير مباشرة. مثال ذلك أن يتم حساب معيار التقييس على كامل البيانات قبل الفصل، ثم استخدامه في الاختبار. الحل هو وضع المعالجة داخل Pipeline بحيث يتعلم كل محول معلماتِه من بيانات التدريب فقط. هذه التفاصيل الصغيرة هي التي تفصل تجربة تعليمية سليمة عن رقم أداء مضلل.

Data leakage occurs when information from the test set enters model training directly or indirectly. For example, scaling the entire dataset before the train/test split allows test information to influence the transformation. A robust solution is to place preprocessing inside a Pipeline so each transformer learns its

parameters from training data only. Such details separate a sound educational experiment from a misleading performance number.

23.3 عدم توازن الفئات

23.3 Class Imbalance

انطباعًا أفضل من الحقيقة. يجب النظر إلى Accuracy عندما تكون إحدى الفئات أكثر عددًا من الأخرى، قد تعطي في الجدول الدوري نفسه قد .stratified split ومصفوفة الالتباس، وربما استخدام F1وRecall وPrecision تكون بعض التصنيفات قليلة العدد، ولذلك ينبغي أن يعرف القارئ أن المقياس يتأثر بتوزيع الفئات وليس بخوارزمية التصنيف وحدها.

When one class is much more common than another, Accuracy can look better than the model really is. Precision, Recall, F1, and the confusion matrix can provide a more informative view, together with a stratified split when appropriate. Some periodic-table categories are small, so the reader should recognize that evaluation depends on class distribution as well as on the classifier.

23.4 التحقق المتقاطع كأداة لتقدير الاستقرار

23.4 Cross-Validation as a Stability Tool

التحقق المتقاطع يقسم البيانات إلى طيات متعددة ويكرر التدريب والتقييم. مع 118 عنصرًا لا تزال العينة صغيرة، لكن التكرار يعطي صورة أفضل من تقسيم واحد. يجب تسجيل المتوسط والانحراف المعياري لكل مقياس، وعدم اختيار النموذج فقط لأن طية واحدة أعطته أعلى نتيجة. الغرض هو تقدير حساسية القرار لعينات التدريب المتاحة.

Cross-validation repeatedly trains and evaluates a model on multiple folds. With only 118 elements the sample remains small, but repeated folds provide more information than a single split. Mean and standard deviation should be recorded for each metric, and a model should not be selected merely because one fold produced the highest score. The purpose is to estimate how sensitive the decision is to the available training samples.

الفصل الرابع والعشرون: هندسة الدقة في الحوسبة الكمومية

Chapter 24: Accuracy Engineering in Quantum Computing

24.1 الفرق بين المحاكاة والعتاد الحقيقي

24.1 Simulation Versus Real Hardware

المحاكاة الكلاسيكية لدائرة كمومية تسمح بتكرار النتائج في بيئة محددة، لكنها لا تعيد كل خصائص العتاد الحقيقي. عند استخدام QPU تظهر الضوضاء، وأخطاء البوابات والقياس، ومحدودية التوصيل، وتكلفة التشغيل. لذلك يجب أن يسجل المشروع backend وعدد shots ونسخة البرنامج وإعدادات transpilation عندما ينتقل من المحاكاة إلى جهاز حقيقي.

A classical simulation of a quantum circuit can be reproducible within a defined environment, but it does not reproduce every property of real hardware. On a QPU, noise, gate and measurement errors, connectivity constraints, and execution cost appear. The project should therefore record the backend, software version, shot count, and transpilation settings whenever it moves from simulation to real hardware.

24.2 قياس الخطأ بدل إخفائه

24.2 Measure Error Instead of Hiding It

لا ينبغي أن تعرض نتيجة الطاقة الكمومية وحدها. يجب أن تعرض أيضًا مرجعًا كلاسيكيًا عندما يكون متاحًا، والفرق المطلق، وإعدادات المحاكاة، وربما تكرار التجربة. إذا تغيرت الطاقة بسبب الضوضاء أو المحسن، فهذا جزء من النتيجة وليس شيئًا يجب حذفه من التقرير. السجل الجيد يحول عدم اليقين من مشكلة مخفية إلى معلومة قابلة للفحص.

A quantum energy result should not be shown alone. When available, a classical reference, absolute difference, simulation settings, and repeated runs should also be reported. If energy changes because of noise or optimizer behavior, that is part of the

result rather than something to remove from the report. Good provenance turns uncertainty from a hidden problem into inspectable information.

24.3 المقارنة مع FCI في مثال H_2

24.3 Comparing H_2 with an FCI Reference

في مثال H_2 يمكن استخدام حل FCI داخل PySCF كمرجع كلاسيكي ضمن نفس الهندسة وقاعدة الأساس. لا يعني التطابق العددي الكامل أن كل شيء صحيح؛ بل يعني أن المسارين اتفقا ضمن تعريف المسألة والتسامح المحدد. وإذا اختلفا، فالمقارنة تقدم سؤالًا تشخيصيًا: هل الاختلاف من ansatz، أو optimizer، أو mapping، أو عدد القياسات، أو إعدادات الحساب المرجعي؟

For H_2, a full configuration interaction calculation in PySCF can provide a classical reference under the same geometry and basis set. Exact numerical agreement is not proof that every component is correct; it means the two workflows agree within the problem definition and chosen tolerance. If they differ, the comparison becomes diagnostic: is the difference due to the ansatz, optimizer, mapping, shot count, or reference calculation settings.

24.4 معنى 4–6 كيوبت في هذا المشروع

24.4 What 4–6 Qubits Mean Here

ذكر 4–6 كيوبت يحدد اتجاهًا تعليميًا وليس وعدًا بأن كل نظام متعدد الإلكترونات يصبح سهلًا. عدد الكيوبتات يتأثر بالتمثيل واختيار الفضاء الفعال، وليس بعدد الإلكترونات وحده. كما أن عدد الدورات وعمق الدائرة يؤثران في الكلفة واحتمال الخطأ. لذلك يجب أن يتبع كل توسع كمومي مواصفة للمشكلة ثم اختبارًا مستقلًا.

The 4–6 qubit range is an educational direction, not a promise that every multi-electron system becomes easy. Qubit count depends on representation and active-space choices, not simply on electron count. Circuit depth and gate count also affect cost and error exposure. Every quantum extension should therefore begin with a problem specification and end with independent validation.

الفصل الخامس والعشرون: الحوكمة والتشغيل المسؤول في الأنظمة العلمية

Chapter 25: Governance and Responsible Operation in Scientific Systems

25.1 الموافقة المسبقة كإشارة تحكم

25.1 Pre-Approval as a Control Signal

عندما ينتقل النظام من تحليل وصفي إلى اقتراحات يمكن أن تؤثر في قرار، تصبح هناك حاجة إلى بوابة مراجعة. يمكن أن تكون الموافقة المسبقة سجلًا يحتوي على هوية المهمة، والبيانات المستخدمة، والإصدار، ونطاق المخرجات، والمراجع المطلوبة. لا تعني البوابة أن كل قرار علمي يحتاج لجنة؛ تعني أن النظام يعرف متى يتجاوز الاستخدام منخفض المخاطر إلى استخدام يحتاج مراجعة إضافية.

When a system moves from descriptive analysis to proposals that may influence a decision, a review gate becomes useful. Pre-approval can be represented as a record containing task identity, data used, version, output scope, and required reviewers. This does not mean every scientific action needs a committee; it means the system can distinguish low-risk exploration from work that needs additional review.

25.2 سجل التدقيق وسلسلة القرار

25.2 Audit Logs and Decision Chains

سجل التدقيق يربط المدخل بالنتيجة والنسخة والشخص أو العملية التي أطلقت الحساب. إذا اقترح النظام مرشحًا، يجب أن يمكن العودة إلى البيانات التي أنتجته والميزات والمعاملات المستخدمة. وعند رفض الاقتراح يجب أن يسجل سبب الرفض عندما يكون ذلك جزءًا من السياسة. هذه البنية تجعل المراجعة لاحقة ممكنة بدل الاعتماد على ذاكرة الفريق.

An audit log links an input to its result, version, and the person or process that initiated the computation. If the system proposes a candidate, it should be possible to trace back to the data, features, and parameters that produced it. If the candidate is rejected, the

reason can be recorded when policy requires it. This makes later review possible without relying on team memory.

الفصل بين النموذج والقرار 25.3

25.3 Separating the Model from the Decision

حتى النموذج عالي الأداء لا يملك سلطة القرار تلقائيًا. يمكن أن يكون النموذج طبقة تقدير، بينما تبقى طبقة القرار مسؤولة عن القيود والسياق والأدلة والمراجعة. هذا الفصل مهم خصوصًا في البيانات الحيوية، حيث قد تكون تكلفة الخطأ عالية. يقرر ما إذا كانت النتيجة مسموحة decision layerينتج نتيجة، و inference :هندسيًا يمكن تمثيله بواجهتين للاستخدام التالي.

Even a high-performing model does not automatically have decision authority. The model can serve as an inference layer, while a separate decision layer applies constraints, context, evidence, and review. This separation is especially important in biological data where the cost of error can be high. Architecturally, inference produces a result and the decision layer determines whether that result is permitted for the next step.

الشفافية ليست كشفًا لكل شيء 25.4

25.4 Transparency Is Not Unlimited Disclosure

الشفافية العلمية تعني أن القارئ يستطيع فهم مصدر النتيجة وحدودها، لكنها لا تعني نشر بيانات حساسة أو أسرار اعتماد أو مفاتيح وصول. لذلك يجمع النظام بين reproducibility وprivacy: يشارك الكود والإصدارات والمقاييس عندما يمكن، ويحجب أو يقلل البيانات التي قد تضر بالأفراد أو المؤسسة. هذا التوازن جزء من هندسة النظام وليس فقرة أخلاقية منفصلة عنه.

Scientific transparency means that a reader can understand where a result came from and what its limits are; it does not require publishing sensitive data, credentials, or protected information. The system therefore balances reproducibility with privacy: share code, versions, and metrics where possible while minimizing or protecting

data that could harm individuals or institutions. This balance is an engineering property, not a detached ethics paragraph.

الفهرسة العملية: ما الذي نكسبه وما الذي ندفعه

Practical Indexing: Benefits and Costs

عمليًا، قرار إنشاء الفهرس يسبقه سؤال عن شكل الاستعلام. إذا كان القارئ يبحث عن المجموعة أو الدورة، فالفهرس منطقي. أما إنشاء فهرس على كل عمود فيضيف كلفة دون فائدة واضحة. هذا المثال يعلّم أن البنية الفيزيائية لقاعدة البيانات يجب أن تتبع الاستخدام المتوقع، مع قياس الأداء بدل الاعتماد على الحدس. على group_number أو period_number

In practice, the decision to create an index begins with the shape of the query. If readers repeatedly filter by group or period, indexes on group_number or period_number are defensible. Indexing every column adds maintenance cost without a clear benefit. The lesson is that physical database structures should follow observed usage patterns and measurements rather than intuition.

يمكن أيضًا أن يكون الفهرس مركبًا عندما يستخدم الاستعلام عمودين معًا. لكن ترتيب الأعمدة داخل الفهرس المركب مهم، لأن المحرك يستفيد أكثر عندما تتوافق البنية مع بداية شروط البحث والترتيب. في مشروع صغير يمكن شرح الفكرة دون ادعاء مكاسب زمنية كبيرة؛ القيمة الحقيقية هي تعلم كيفية التفكير في خطة الوصول إلى البيانات.

A composite index can be useful when queries repeatedly use two columns together. The order of columns inside the composite index matters because the engine benefits most when the structure matches the leading search and ordering conditions. On this small project, the purpose is not to claim dramatic speedups but to teach how to reason about data-access paths.

الفهرس نفسه يحتاج إلى تحقق. إذا كان الاستعلام يستخدمه، ينبغي أن يظهر ذلك في خطة التنفيذ أو في قياس مقارن على حجم بيانات مناسب. وإذا لم يستخدمه المحرك، يجب أن نعرف لماذا قبل إضافة فهرس آخر. بهذه الطريقة يصبح تحسين الأداء دورة: فرضية، قياس، تعديل، قياس جديد.

The index itself should be validated. If a query is expected to use it, the execution plan or an appropriate benchmark should provide evidence. If the engine does not use it, the reason should be understood before another index is added. Performance engineering is therefore a cycle of hypothesis, measurement, change, and re-measurement.

في قواعد البيانات العلمية الأكبر قد تتغير طبيعة المشكلة عندما تصبح البيانات موزعة أو عندما تتكرر عمليات التحميل. عندها تظهر مفاهيم مثل partition pruning أو bulk inserts أو التخزين العمودي. لا يحتاج القارئ إلى تطبيقها على 118 عنصرًا، لكنه يحتاج إلى فهم أن الاختيار المعماري تابع لحجم البيانات ونمط الوصول.

In larger scientific databases the problem can change when data become distributed or loading becomes repetitive. Concepts such as partition pruning, bulk inserts, or columnar storage may then become relevant. They do not need to be implemented for 118 elements, but the reader should understand that architecture follows data volume and access patterns.

المبدأ النهائي هو عدم الخلط بين تعقيد النظام وجودته. قاعدة بسيطة بفهرسين مناسبين قد تكون أفضل من قاعدة مليئة بالفهارس غير المستخدمة. الهندسة الجيدة تزيل التعقيد الذي لا يخدم سؤالًا أو قيدًا أو قياسًا محددًا.

The final principle is not to confuse system complexity with quality. A simple database with two useful indexes can be better than a database filled with unused indexes. Good engineering removes complexity that does not serve a defined question, constraint, or measurement.

قراءة خطة التنفيذ

Reading the Execution Plan

خطة التنفيذ يمكن أن تكشف فرقًا مهمًا بين الاستعلام النظري والاستعلام الذي يراه المحرك فعليًا. قد يكتب القارئ شرطًا يبدو انتقائيًا، لكن التحويل أو نوع البيانات أو غياب الفهرس يجعل المحرك يقرأ عددًا كبيرًا من الصفوف. لذلك يجب أن يتعلم القارئ قراءة الخطة بوصفها وصفًا لمسار الحساب، لا كإجابة بنعم أو لا.

An execution plan can expose an important difference between the query as imagined and the query path chosen by the engine. A predicate may look selective, yet data types, transformations, or missing indexes can cause many rows to be read. The reader should learn to interpret a plan as a description of computation, not as a simple yes-or-no verdict.

من المفيد الاحتفاظ بالاستعلام الأصلي إلى جانب الخطة، لأن الخطة وحدها لا تخبرنا لماذا كتبنا الاستعلام. في مشروع تعليمي، هذا الربط يعلّم توثيق القرار: السؤال العلمي، الاستعلام، خطة التنفيذ، ثم النتيجة.

It is useful to keep the original query next to its execution plan because the plan alone does not explain why the query exists. In an educational project this creates a decision record: scientific question, SQL query, execution plan, and result.

عند مقارنة خطتين يجب أن يكون حجم البيانات وظروف التنفيذ متقاربة. تشغيل استعلام مرة واحدة على جهاز مشغول لا يكفي للحكم على فرق صغير في الزمن. لهذا نفضل في الكتاب تفسيرًا بنيويًا للخطة، ونستخدم القياس عندما تكون البيئة قابلة للضبط.

When comparing two plans, data volume and execution conditions should be comparable. Running a query once on a busy machine is not enough to establish a small timing difference. The book therefore emphasizes structural interpretation of plans and uses benchmarks when the environment can be controlled.

هذا الأسلوب يربط SQL بالمنهج العلمي نفسه: لا نقول إن التصميم أسرع لأننا نفضله؛ نقول إن القياس أو خطة التنفيذ يدعمان الادعاء ضمن شروط التجربة.

This approach connects SQL to the scientific method itself: we do not say a design is faster because we prefer it; we say that a measurement or execution plan supports the claim under defined conditions.

بذلك يصبح تغير الأداء provenance. وعندما يكبر النظام، يمكن حفظ خطط التنفيذ أو ملخصات الأداء ضمن سجل عبر إصدارات قاعدة البيانات أو البيانات قابلًا للمراجعة، بدل أن يختفي في ذاكرة المطور.

As the system grows, execution plans or performance summaries can be stored as provenance artifacts. Performance changes across database or data versions then become reviewable instead of depending on developer memory.

سلامة المعاملات والتحقق

Transaction Safety and Validation

المعاملة تحدد وحدة الاتساق، لكنها لا تتحقق من المعنى العلمي. يمكن لمعاملة أن تحفظ صفًا خاطئًا بصورة مثالية من الناحية التقنية. لذلك يبقى التحقق بعد الإدخال ضروريًا، مثل عد العناصر، وفحص تفرد الرموز، ومراجعة القيم خارج النطاق.

A transaction defines a consistency unit, but it does not validate scientific meaning. A transaction can perfectly preserve an incorrect row. Post-load validation therefore remains essential: count records, check symbol uniqueness, and detect values outside expected ranges.

عند فشل جزء من عملية الإدخال، يجب rollback في منع حالة نصف مكتملة. لكن إذا كانت العملية طويلة، تكون قيمة التفكير في حجم المعاملة والذاكرة وزمن القفل. في هذا المثال يمكن تحميل 118 عنصرًا في معاملة واحدة بسهولة، بينما قد تحتاج قواعد ضخمة إلى دفعات مدروسة.

When part of a load fails, rollback prevents a partially committed state. For long operations, however, transaction size, memory, and lock duration become engineering concerns. One transaction is convenient for 118 elements, while very large databases may need carefully designed batches.

التحقق من القيود يجب أن يشمل foreign keys أيضًا. إذا أُدخل عنصر يشير إلى مجموعة غير موجودة، فالمشكلة ليست في النص الإنجليزي أو العربي بل في العلاقة. هذا يوضح لماذا نحتاج إلى قيود على مستوى قاعدة البيانات لا إلى قواعد في واجهة المستخدم فقط.

Validation should include foreign keys. If an element points to a nonexistent group, the problem is relational rather than linguistic.

This illustrates why constraints belong in the database itself rather than only in a user interface.

في نظام قابل للتدقيق، يجب أن يسجل الحدث وقت التنفيذ والإصدار وعدد الصفوف المتأثرة. هذه المعلومات تجعل إعادة الإنتاج ممكنة حتى إذا تغيرت البيئة لاحقًا.

In an auditable system, execution time, software version, and affected-row counts should be recorded. These details support reproduction even if the environment changes later.

بهذا يصبح مفهوم transaction جزءًا من هندسة الثقة: لا يحل كل مشكلة، لكنه يمنع فئة محددة من أخطاء الحالة الوسيطة ويجعل حدود التغيير واضحة.

Transaction design therefore becomes part of trust engineering: it does not solve every problem, but it prevents a defined class of intermediate-state failures and makes change boundaries explicit.

التوسع من حيث الحاجة

Scaling by Need

الانتقال إلى نظام أكبر يبدأ بتحديد ما إذا كانت البيانات نفسها هي المشكلة. إذا كانت 90% من وقت التنفيذ تذهب إلى استعلام واحد، فإعادة كتابة الاستعلام قد تكون أهم من إضافة Spark. وإذا كانت البيانات أكبر من الذاكرة، قد يكون Dask أو تخزين مجزأ أكثر منطقية.

Scaling begins by identifying whether the data volume is actually the problem. If one query consumes most of the runtime, rewriting it may matter more than adding Spark. If data no longer fit in memory, Dask or partitioned storage may be more appropriate.

يجب أن يظل SQL حاضرًا في الطبقة العلائقية حتى عند إضافة أدوات موزعة عندما يكون ذلك مناسبًا. توزيع المعالجة لا يلغي الحاجة إلى تعريف المفاتيح والعلاقات والقيود.

SQL can remain valuable in the relational layer even when distributed tools are added. Distributed computation does not remove the need for keys, relationships, and integrity constraints.

التوسع الأفقي يضيف كلفة في التنسيق والمراقبة والفشل الجزئي. لذلك لا ينبغي أن يكون الهدف هو أكبر بنية ممكنة، بل البنية التي تحقق متطلبات الأداء والموثوقية بأقل تعقيد معقول.

Horizontal scaling adds coordination, monitoring, and partial-failure costs. The goal should therefore not be the largest possible architecture, but the architecture that meets performance and reliability requirements with reasonable complexity.

هذا التفكير يفسر لماذا يبدأ المشروع بقاعدة SQLite صغيرة. القارئ يرى النموذج المنطقي أولًا، ثم يمكنه نقل النموذج إلى PostgreSQL أو مستودع موزع عندما تتغير القيود.

This is why the project starts with a small SQLite database. The reader learns the logical model first and can later move it to PostgreSQL or a distributed store when constraints change.

النقطة الأساسية أن التوسع ليس فصلًا عن التصميم الأصلي؛ هو اختبار له. إذا كانت العلاقات واضحة والبيانات موثقة، يصبح النقل إلى بنية أكبر أسهل وأقل مخاطرة.

Scaling is not separate from the original design; it is a test of it. Clear relationships and documented data make migration to larger architectures easier and less risky.

تعريف الميزة ومصدرها

Feature Definition and Provenance

في هذا القسم، نعامل الميزات كجزء من تصميم النظام وليس كإضافة تجميلية. يجب أن يكون له تعريف واضح، ومدخلات قابلة للتتبع، ومخرج يمكن اختباره. إذا لم نستطع تحديد ما يدخل إلى العملية وما يخرج منها، فلن نستطيع تفسير نتيجة تغيرها بين تشغيل وآخر.

In this section, features is treated as part of system design rather than as decoration. It needs a clear definition, traceable inputs, and testable outputs. If we cannot state what enters the process and what leaves it, we cannot explain why a result changes between runs.

الخطوة العملية هي تسجيل القرار إلى جانب النتيجة. لا يكفي أن نحتفظ بالرقم النهائي؛ نحتاج إلى معرفة الإصدار والبيانات والمعاملات التي أنتجته. هذا السجل يحمي المشروع من مشكلة إعادة الإنتاج التي تظهر عندما يعمل المثال مرة ثم يفشل بعد تحديث بيئة أو مصدر بيانات.

The practical step is to record the decision together with the result. Keeping only the final number is insufficient; we need the version, data, and parameters that produced it. This protects the project from reproducibility failures that appear when an example works once and changes after an environment or data-source update.

هناك أيضًا فرق بين الاختبار والتحقق. الاختبار يسأل هل الكود يحقق عقده البرمجي، بينما التحقق العلمي يسأل هل الناتج يجيب عن السؤال المحدد بطريقة معقولة. الجمع بين الاثنين يمنع نجاحًا برمجيًا يخفي خطأً في تفسير المسألة.

There is also a distinction between testing and validation. A software test asks whether code satisfies its contract, while scientific validation asks whether the result answers the stated question appropriately. Combining both prevents a software success from hiding a scientific interpretation error.

عند ظهور نتيجة غير متوقعة، يجب أن يكون المسار التشخيصي مرتبًا: تحقق من المدخلات، ثم التحويلات، ثم الخوارزمية، ثم المقياس، ثم المصدر المرجعي. القفز مباشرة إلى تغيير النموذج يجعل من الصعب معرفة سبب التحسن أو التدهور.

When an unexpected result appears, diagnosis should follow an ordered path: inputs, transformations, algorithm, metric, and reference source. Jumping directly to a new model makes it difficult to know why performance improved or deteriorated.

وأخيرًا، ينبغي أن يبقى هذا القسم مرتبطًا ببقية المشروع. النتيجة ليست نهاية الخط؛ يمكن أن تصبح سجلًا في SQL، أو مدخلًا لتقرير، أو سببًا لطلب مراجعة، أو إشارة إلى أن المرحلة التالية غير مسموحة بعد. بهذه الطريقة يتحول المفهوم إلى جزء من نظام متكامل.

Finally, this section should remain connected to the rest of the project. A result is not necessarily the end of the pipeline; it can become a SQL record, a report input, a reason for review, or a signal

that the next stage is not yet permitted. The concept thus becomes part of an integrated system.

منع التسرب في خط المعالجة

Leakage Prevention in the Pipeline

في هذا القسم، نعامل تسرب البيانات كجزء من تصميم النظام وليس كإضافة تجميلية. يجب أن يكون له تعريف واضح، ومدخلات قابلة للتتبع، ومخرج يمكن اختباره. إذا لم نستطع تحديد ما يدخل إلى العملية وما يخرج منها، فلن نستطيع تفسير نتيجة تغيرها بين تشغيل وآخر.

In this section, leakage is treated as part of system design rather than as decoration. It needs a clear definition, traceable inputs, and testable outputs. If we cannot state what enters the process and what leaves it, we cannot explain why a result changes between runs.

الخطوة العملية هي تسجيل القرار إلى جانب النتيجة. لا يكفي أن نحتفظ بالرقم النهائي؛ نحتاج إلى معرفة الإصدار والبيانات والمعاملات التي أنتجته. هذا السجل يحمي المشروع من مشكلة إعادة الإنتاج التي تظهر عندما يعمل المثال مرة ثم يفشل بعد تحديث بيئة أو مصدر بيانات.

The practical step is to record the decision together with the result. Keeping only the final number is insufficient; we need the version, data, and parameters that produced it. This protects the project from reproducibility failures that appear when an example works once and changes after an environment or data-source update.

هناك أيضًا فرق بين الاختبار والتحقق. الاختبار يسأل هل الكود يحقق عقده البرمجي، بينما التحقق العلمي يسأل هل الناتج يجيب عن السؤال المحدد بطريقة معقولة. الجمع بين الاثنين يمنع نجاحًا برمجيًا يخفي خطأً في تفسير المسألة.

There is also a distinction between testing and validation. A software test asks whether code satisfies its contract, while scientific validation asks whether the result answers the stated question appropriately. Combining both prevents a software success from hiding a scientific interpretation error.

عند ظهور نتيجة غير متوقعة، يجب أن يكون المسار التشخيصي مرتبًا: تحقق من المدخلات، ثم التحويلات، ثم الخوارزمية، ثم المقياس، ثم المصدر المرجعي. القفز مباشرة إلى تغيير النموذج يجعل من الصعب معرفة سبب التحسن أو التدهور.

When an unexpected result appears, diagnosis should follow an ordered path: inputs, transformations, algorithm, metric, and reference source. Jumping directly to a new model makes it difficult to know why performance improved or deteriorated.

وأخيرًا، ينبغي أن يبقى هذا القسم مرتبطًا ببقية المشروع. النتيجة ليست نهاية الخط؛ يمكن أن تصبح سجلًا في SQL، أو مدخلًا لتقرير، أو سببًا لطلب مراجعة، أو إشارة إلى أن المرحلة التالية غير مسموحة بعد. بهذه الطريقة يتحول المفهوم إلى جزء من نظام متكامل.

Finally, this section should remain connected to the rest of the project. A result is not necessarily the end of the pipeline; it can become a SQL record, a report input, a reason for review, or a signal that the next stage is not yet permitted. The concept thus becomes part of an integrated system.

قراءة مقاييس التصنيف

Interpreting Classification Metrics

في هذا القسم، نعامل عدم توازن الفئات كجزء من تصميم النظام وليس كإضافة تجميلية. يجب أن يكون له تعريف واضح، ومدخلات قابلة للتتبع، ومخرج يمكن اختباره. إذا لم نستطع تحديد ما يدخل إلى العملية وما يخرج منها، فلن نستطيع تفسير نتيجة تغيرها بين تشغيل وآخر.

In this section, imbalance is treated as part of system design rather than as decoration. It needs a clear definition, traceable inputs, and testable outputs. If we cannot state what enters the process and what leaves it, we cannot explain why a result changes between runs.

الخطوة العملية هي تسجيل القرار إلى جانب النتيجة. لا يكفي أن نحتفظ بالرقم النهائي؛ نحتاج إلى معرفة الإصدار والبيانات والمعاملات التي أنتجته. هذا السجل يحمي المشروع من مشكلة إعادة الإنتاج التي تظهر عندما يعمل المثال مرة ثم يفشل بعد تحديث بيئة أو مصدر بيانات.

The practical step is to record the decision together with the result. Keeping only the final number is insufficient; we need the version, data, and parameters that produced it. This protects the project from reproducibility failures that appear when an example works once and changes after an environment or data-source update.

هناك أيضًا فرق بين الاختبار والتحقق. الاختبار يسأل هل الكود يحقق عقده البرمجي، بينما التحقق العلمي يسأل هل الناتج يجيب عن السؤال المحدد بطريقة معقولة. الجمع بين الاثنين يمنع نجاحًا برمجيًا يخفي خطأً في تفسير المسألة.

There is also a distinction between testing and validation. A software test asks whether code satisfies its contract, while scientific validation asks whether the result answers the stated question appropriately. Combining both prevents a software success from hiding a scientific interpretation error.

عند ظهور نتيجة غير متوقعة، يجب أن يكون المسار التشخيصي مرتبًا: تحقق من المدخلات، ثم التحويلات، ثم الخوارزمية، ثم المقياس، ثم المصدر المرجعي. القفز مباشرة إلى تغيير النموذج يجعل من الصعب معرفة سبب التحسن أو التدهور.

When an unexpected result appears, diagnosis should follow an ordered path: inputs, transformations, algorithm, metric, and reference source. Jumping directly to a new model makes it difficult to know why performance improved or deteriorated.

وأخيرًا، ينبغي أن يبقى هذا القسم مرتبطًا ببقية المشروع. النتيجة ليست نهاية الخط؛ يمكن أن تصبح سجلًا في SQL، أو مدخلًا لتقرير، أو سببًا لطلب مراجعة، أو إشارة إلى أن المرحلة التالية غير مسموحة بعد. بهذه الطريقة يتحول المفهوم إلى جزء من نظام متكامل.

Finally, this section should remain connected to the rest of the project. A result is not necessarily the end of the pipeline; it can become a SQL record, a report input, a reason for review, or a signal that the next stage is not yet permitted. The concept thus becomes part of an integrated system.

تفسير نتائج التحقق المتقاطع

Interpreting Cross-Validation Results

في هذا القسم، نعامل التحقق المتقاطع كجزء من تصميم النظام وليس كإضافة تجميلية. يجب أن يكون له تعريف واضح، ومدخلات قابلة للتتبع، ومخرج يمكن اختباره. إذا لم نستطع تحديد ما يدخل إلى العملية وما يخرج منها، فلن نستطيع تفسير نتيجة تغيرها بين تشغيل وآخر.

In this section, cross-validation is treated as part of system design rather than as decoration. It needs a clear definition, traceable inputs, and testable outputs. If we cannot state what enters the process and what leaves it, we cannot explain why a result changes between runs.

الخطوة العملية هي تسجيل القرار إلى جانب النتيجة. لا يكفي أن نحتفظ بالرقم النهائي؛ نحتاج إلى معرفة الإصدار والبيانات والمعاملات التي أنتجته. هذا السجل يحمي المشروع من مشكلة إعادة الإنتاج التي تظهر عندما يعمل المثال مرة ثم يفشل بعد تحديث بيئة أو مصدر بيانات.

The practical step is to record the decision together with the result. Keeping only the final number is insufficient; we need the version, data, and parameters that produced it. This protects the project from reproducibility failures that appear when an example works once and changes after an environment or data-source update.

هناك أيضًا فرق بين الاختبار والتحقق. الاختبار يسأل هل الكود يحقق عقده البرمجي، بينما التحقق العلمي يسأل هل الناتج يجيب عن السؤال المحدد بطريقة معقولة. الجمع بين الاثنين يمنع نجاحًا برمجيًا يخفي خطأً في تفسير المسألة.

There is also a distinction between testing and validation. A software test asks whether code satisfies its contract, while scientific validation asks whether the result answers the stated question appropriately. Combining both prevents a software success from hiding a scientific interpretation error.

عند ظهور نتيجة غير متوقعة، يجب أن يكون المسار التشخيصي مرتبًا: تحقق من المدخلات، ثم التحويلات، ثم الخوارزمية، ثم المقياس، ثم المصدر المرجعي. القفز مباشرة إلى تغيير النموذج يجعل من الصعب معرفة سبب التحسن أو التدهور.

When an unexpected result appears, diagnosis should follow an ordered path: inputs, transformations, algorithm, metric, and reference source. Jumping directly to a new model makes it difficult to know why performance improved or deteriorated.

وأخيرًا، ينبغي أن يبقى هذا القسم مرتبطًا ببقية المشروع. النتيجة ليست نهاية الخط؛ يمكن أن تصبح سجلًا في SQL، أو مدخلًا لتقرير، أو سببًا لطلب مراجعة، أو إشارة إلى أن المرحلة التالية غير مسموحة بعد. بهذه الطريقة يتحول المفهوم إلى جزء من نظام متكامل.

Finally, this section should remain connected to the rest of the project. A result is not necessarily the end of the pipeline; it can become a SQL record, a report input, a reason for review, or a signal that the next stage is not yet permitted. The concept thus becomes part of an integrated system.

سجل الانتقال من المحاكاة إلى العتاد

Recording the Move from Simulation to Hardware

في هذا القسم، نعامل العتاد الكمومي كجزء من تصميم النظام وليس كإضافة تجميلية. يجب أن يكون له تعريف واضح، ومدخلات قابلة للتتبع، ومخرج يمكن اختباره. إذا لم نستطع تحديد ما يدخل إلى العملية وما يخرج منها، فلن نستطيع تفسير نتيجة تغيرها بين تشغيل وآخر.

In this section, hardware is treated as part of system design rather than as decoration. It needs a clear definition, traceable inputs, and testable outputs. If we cannot state what enters the process and what leaves it, we cannot explain why a result changes between runs.

الخطوة العملية هي تسجيل القرار إلى جانب النتيجة. لا يكفي أن نحتفظ بالرقم النهائي؛ نحتاج إلى معرفة الإصدار والبيانات والمعاملات التي أنتجته. هذا السجل يحمي المشروع من مشكلة إعادة الإنتاج التي تظهر عندما يعمل المثال مرة ثم يفشل بعد تحديث بيئة أو مصدر بيانات.

The practical step is to record the decision together with the result. Keeping only the final number is insufficient; we need the version,

data, and parameters that produced it. This protects the project from reproducibility failures that appear when an example works once and changes after an environment or data-source update.

هناك أيضًا فرق بين الاختبار والتحقق. الاختبار يسأل هل الكود يحقق عقده البرمجي، بينما التحقق العلمي يسأل هل الناتج يجيب عن السؤال المحدد بطريقة معقولة. الجمع بين الاثنين يمنع نجاحًا برمجيًا يخفي خطأً في تفسير المسألة.

There is also a distinction between testing and validation. A software test asks whether code satisfies its contract, while scientific validation asks whether the result answers the stated question appropriately. Combining both prevents a software success from hiding a scientific interpretation error.

عند ظهور نتيجة غير متوقعة، يجب أن يكون المسار التشخيصي مرتبًا: تحقق من المدخلات، ثم التحويلات، ثم الخوارزمية، ثم المقياس، ثم المصدر المرجعي. القفز مباشرة إلى تغيير النموذج يجعل من الصعب معرفة سبب التحسن أو التدهور.

When an unexpected result appears, diagnosis should follow an ordered path: inputs, transformations, algorithm, metric, and reference source. Jumping directly to a new model makes it difficult to know why performance improved or deteriorated.

وأخيرًا، ينبغي أن يبقى هذا القسم مرتبطًا ببقية المشروع. النتيجة ليست نهاية الخط؛ يمكن أن تصبح سجلًا في SQL، أو مدخلًا لتقرير، أو سببًا لطلب مراجعة، أو إشارة إلى أن المرحلة التالية غير مسموحة بعد. بهذه الطريقة يتحول المفهوم إلى جزء من نظام متكامل.

Finally, this section should remain connected to the rest of the project. A result is not necessarily the end of the pipeline; it can become a SQL record, a report input, a reason for review, or a signal that the next stage is not yet permitted. The concept thus becomes part of an integrated system.

تقدير الخطأ وعدم اليقين

Error and Uncertainty Estimation

في هذا القسم، نعامل الخطأ الكمومي كجزء من تصميم النظام وليس كإضافة تجميلية. يجب أن يكون له تعريف واضح، ومدخلات قابلة للتتبع، ومخرج يمكن اختباره. إذا لم نستطع تحديد ما يدخل إلى العملية وما يخرج منها، فلن نستطيع تفسير نتيجة تغيرها بين تشغيل وآخر.

In this section, error is treated as part of system design rather than as decoration. It needs a clear definition, traceable inputs, and testable outputs. If we cannot state what enters the process and what leaves it, we cannot explain why a result changes between runs.

الخطوة العملية هي تسجيل القرار إلى جانب النتيجة. لا يكفي أن نحتفظ بالرقم النهائي؛ نحتاج إلى معرفة الإصدار والبيانات والمعاملات التي أنتجته. هذا السجل يحمي المشروع من مشكلة إعادة الإنتاج التي تظهر عندما يعمل المثال مرة ثم يفشل بعد تحديث بيئة أو مصدر بيانات.

The practical step is to record the decision together with the result. Keeping only the final number is insufficient; we need the version, data, and parameters that produced it. This protects the project from reproducibility failures that appear when an example works once and changes after an environment or data-source update.

هناك أيضًا فرق بين الاختبار والتحقق. الاختبار يسأل هل الكود يحقق عقده البرمجي، بينما التحقق العلمي يسأل هل الناتج يجيب عن السؤال المحدد بطريقة معقولة. الجمع بين الاثنين يمنع نجاحًا برمجيًا يخفي خطأً في تفسير المسألة.

There is also a distinction between testing and validation. A software test asks whether code satisfies its contract, while scientific validation asks whether the result answers the stated question appropriately. Combining both prevents a software success from hiding a scientific interpretation error.

عند ظهور نتيجة غير متوقعة، يجب أن يكون المسار التشخيصي مرتبًا: تحقق من المدخلات، ثم التحويلات، ثم الخوارزمية، ثم المقياس، ثم المصدر المرجعي. القفز مباشرة إلى تغيير النموذج يجعل من الصعب معرفة سبب التحسن أو التدهور.

When an unexpected result appears, diagnosis should follow an ordered path: inputs, transformations, algorithm, metric, and reference source. Jumping directly to a new model makes it difficult to know why performance improved or deteriorated.

وأخيرًا، ينبغي أن يبقى هذا القسم مرتبطًا ببقية المشروع. النتيجة ليست نهاية الخط؛ يمكن أن تصبح سجلًا في SQL، أو مدخلًا لتقرير، أو سببًا لطلب مراجعة، أو إشارة إلى أن المرحلة التالية غير مسموحة بعد. بهذه الطريقة يتحول المفهوم إلى جزء من نظام متكامل.

Finally, this section should remain connected to the rest of the project. A result is not necessarily the end of the pipeline; it can become a SQL record, a report input, a reason for review, or a signal that the next stage is not yet permitted. The concept thus becomes part of an integrated system.

مقارنة VQE بمرجع مستقل

Comparing VQE with an Independent Reference

في هذا القسم، نعامل مرجع FCI كجزء من تصميم النظام وليس كإضافة تجميلية. يجب أن يكون له تعريف واضح، ومدخلات قابلة للتتبع، ومخرج يمكن اختباره. إذا لم نستطع تحديد ما يدخل إلى العملية وما يخرج منها، فلن نستطيع تفسير نتيجة تغيرها بين تشغيل وآخر.

In this section, FCI is treated as part of system design rather than as decoration. It needs a clear definition, traceable inputs, and testable outputs. If we cannot state what enters the process and what leaves it, we cannot explain why a result changes between runs.

الخطوة العملية هي تسجيل القرار إلى جانب النتيجة. لا يكفي أن نحتفظ بالرقم النهائي؛ نحتاج إلى معرفة الإصدار والبيانات والمعاملات التي أنتجته. هذا السجل يحمي المشروع من مشكلة إعادة الإنتاج التي تظهر عندما يعمل المثال مرة ثم يفشل بعد تحديث بيئة أو مصدر بيانات.

The practical step is to record the decision together with the result. Keeping only the final number is insufficient; we need the version, data, and parameters that produced it. This protects the project

from reproducibility failures that appear when an example works once and changes after an environment or data-source update.

هناك أيضًا فرق بين الاختبار والتحقق. الاختبار يسأل هل الكود يحقق عقده البرمجي، بينما التحقق العلمي يسأل هل الناتج يجيب عن السؤال المحدد بطريقة معقولة. الجمع بين الاثنين يمنع نجاحًا برمجيًا يخفي خطأً في تفسير المسألة.

There is also a distinction between testing and validation. A software test asks whether code satisfies its contract, while scientific validation asks whether the result answers the stated question appropriately. Combining both prevents a software success from hiding a scientific interpretation error.

عند ظهور نتيجة غير متوقعة، يجب أن يكون المسار التشخيصي مرتبًا: تحقق من المدخلات، ثم التحويلات، ثم الخوارزمية، ثم المقياس، ثم المصدر المرجعي. القفز مباشرة إلى تغيير النموذج يجعل من الصعب معرفة سبب التحسن أو التدهور.

When an unexpected result appears, diagnosis should follow an ordered path: inputs, transformations, algorithm, metric, and reference source. Jumping directly to a new model makes it difficult to know why performance improved or deteriorated.

وأخيرًا، ينبغي أن يبقى هذا القسم مرتبطًا ببقية المشروع. النتيجة ليست نهاية الخط؛ يمكن أن تصبح سجلًا في SQL، أو مدخلًا لتقرير، أو سببًا لطلب مراجعة، أو إشارة إلى أن المرحلة التالية غير مسموحة بعد. بهذه الطريقة يتحول المفهوم إلى جزء من نظام متكامل.

Finally, this section should remain connected to the rest of the project. A result is not necessarily the end of the pipeline; it can become a SQL record, a report input, a reason for review, or a signal that the next stage is not yet permitted. The concept thus becomes part of an integrated system.

حدود التوسع بالكيوبتات

Limits of Qubit Scaling

في هذا القسم، نعامل الكيوبتات كجزء من تصميم النظام وليس كإضافة تجميلية. يجب أن يكون له تعريف واضح، ومدخلات قابلة للتتبع، ومخرج يمكن اختباره. إذا لم نستطع تحديد ما يدخل إلى العملية وما يخرج منها، فلن نستطيع تفسير نتيجة تغيرها بين تشغيل وآخر.

In this section, qubits is treated as part of system design rather than as decoration. It needs a clear definition, traceable inputs, and testable outputs. If we cannot state what enters the process and what leaves it, we cannot explain why a result changes between runs.

الخطوة العملية هي تسجيل القرار إلى جانب النتيجة. لا يكفي أن نحتفظ بالرقم النهائي؛ نحتاج إلى معرفة الإصدار والبيانات والمعاملات التي أنتجته. هذا السجل يحمي المشروع من مشكلة إعادة الإنتاج التي تظهر عندما يعمل المثال مرة ثم يفشل بعد تحديث بيئة أو مصدر بيانات.

The practical step is to record the decision together with the result. Keeping only the final number is insufficient; we need the version, data, and parameters that produced it. This protects the project from reproducibility failures that appear when an example works once and changes after an environment or data-source update.

هناك أيضًا فرق بين الاختبار والتحقق. الاختبار يسأل هل الكود يحقق عقده البرمجي، بينما التحقق العلمي يسأل هل الناتج يجيب عن السؤال المحدد بطريقة معقولة. الجمع بين الاثنين يمنع نجاحًا برمجيًا يخفي خطأً في تفسير المسألة.

There is also a distinction between testing and validation. A software test asks whether code satisfies its contract, while scientific validation asks whether the result answers the stated question appropriately. Combining both prevents a software success from hiding a scientific interpretation error.

عند ظهور نتيجة غير متوقعة، يجب أن يكون المسار التشخيصي مرتبًا: تحقق من المدخلات، ثم التحويلات، ثم الخوارزمية، ثم المقياس، ثم المصدر المرجعي. القفز مباشرة إلى تغيير النموذج يجعل من الصعب معرفة سبب التحسن أو التدهور.

When an unexpected result appears, diagnosis should follow an ordered path: inputs, transformations, algorithm, metric, and reference source. Jumping directly to a new model makes it difficult to know why performance improved or deteriorated.

أو SQL، وأخيرًا، ينبغي أن يبقى هذا القسم مرتبطًا ببقية المشروع. النتيجة ليست نهاية الخط؛ يمكن أن تصبح سجلًا في مدخلًا لتقرير، أو سببًا لطلب مراجعة، أو إشارة إلى أن المرحلة التالية غير مسموحة بعد. بهذه الطريقة يتحول المفهوم إلى جزء من نظام متكامل.

Finally, this section should remain connected to the rest of the project. A result is not necessarily the end of the pipeline; it can become a SQL record, a report input, a reason for review, or a signal that the next stage is not yet permitted. The concept thus becomes part of an integrated system.

بوابة الموافقة قبل التنفيذ

Pre-Execution Approval Gate

في هذا القسم، نعامل الموافقة المسبقة كجزء من تصميم النظام وليس كإضافة تجميلية. يجب أن يكون له تعريف واضح، ومدخلات قابلة للتتبع، ومخرج يمكن اختباره. إذا لم نستطع تحديد ما يدخل إلى العملية وما يخرج منها، فلن نستطيع تفسير نتيجة تغيرها بين تشغيل وآخر.

In this section, approval is treated as part of system design rather than as decoration. It needs a clear definition, traceable inputs, and testable outputs. If we cannot state what enters the process and what leaves it, we cannot explain why a result changes between runs.

الخطوة العملية هي تسجيل القرار إلى جانب النتيجة. لا يكفي أن نحتفظ بالرقم النهائي؛ نحتاج إلى معرفة الإصدار والبيانات والمعاملات التي أنتجته. هذا السجل يحمي المشروع من مشكلة إعادة الإنتاج التي تظهر عندما يعمل المثال مرة ثم يفشل بعد تحديث بيئة أو مصدر بيانات.

The practical step is to record the decision together with the result. Keeping only the final number is insufficient; we need the version,

data, and parameters that produced it. This protects the project from reproducibility failures that appear when an example works once and changes after an environment or data-source update.

هناك أيضًا فرق بين الاختبار والتحقق. الاختبار يسأل هل الكود يحقق عقده البرمجي، بينما التحقق العلمي يسأل هل الناتج يجيب عن السؤال المحدد بطريقة معقولة. الجمع بين الاثنين يمنع نجاحًا برمجيًا يخفي خطأً في تفسير المسألة.

There is also a distinction between testing and validation. A software test asks whether code satisfies its contract, while scientific validation asks whether the result answers the stated question appropriately. Combining both prevents a software success from hiding a scientific interpretation error.

عند ظهور نتيجة غير متوقعة، يجب أن يكون المسار التشخيصي مرتبًا: تحقق من المدخلات، ثم التحويلات، ثم الخوارزمية، ثم المقياس، ثم المصدر المرجعي. القفز مباشرة إلى تغيير النموذج يجعل من الصعب معرفة سبب التحسن أو التدهور.

When an unexpected result appears, diagnosis should follow an ordered path: inputs, transformations, algorithm, metric, and reference source. Jumping directly to a new model makes it difficult to know why performance improved or deteriorated.

وأخيرًا، ينبغي أن يبقى هذا القسم مرتبطًا ببقية المشروع. النتيجة ليست نهاية الخط؛ يمكن أن تصبح سجلًا في SQL، أو مدخلًا لتقرير، أو سببًا لطلب مراجعة، أو إشارة إلى أن المرحلة التالية غير مسموحة بعد. بهذه الطريقة يتحول المفهوم إلى جزء من نظام متكامل.

Finally, this section should remain connected to the rest of the project. A result is not necessarily the end of the pipeline; it can become a SQL record, a report input, a reason for review, or a signal that the next stage is not yet permitted. The concept thus becomes part of an integrated system.

تتبع سلسلة القرار

Tracing the Decision Chain

في هذا القسم، نعامل سجل التدقيق كجزء من تصميم النظام وليس كإضافة تجميلية. يجب أن يكون له تعريف واضح، ومدخلات قابلة للتتبع، ومخرج يمكن اختباره. إذا لم نستطع تحديد ما يدخل إلى العملية وما يخرج منها، فلن نستطيع تفسير نتيجة تغيرها بين تشغيل وآخر.

In this section, audit is treated as part of system design rather than as decoration. It needs a clear definition, traceable inputs, and testable outputs. If we cannot state what enters the process and what leaves it, we cannot explain why a result changes between runs.

الخطوة العملية هي تسجيل القرار إلى جانب النتيجة. لا يكفي أن نحتفظ بالرقم النهائي؛ نحتاج إلى معرفة الإصدار والبيانات والمعاملات التي أنتجته. هذا السجل يحمي المشروع من مشكلة إعادة الإنتاج التي تظهر عندما يعمل المثال مرة ثم يفشل بعد تحديث بيئة أو مصدر بيانات.

The practical step is to record the decision together with the result. Keeping only the final number is insufficient; we need the version, data, and parameters that produced it. This protects the project from reproducibility failures that appear when an example works once and changes after an environment or data-source update.

هناك أيضًا فرق بين الاختبار والتحقق. الاختبار يسأل هل الكود يحقق عقده البرمجي، بينما التحقق العلمي يسأل هل الناتج يجيب عن السؤال المحدد بطريقة معقولة. الجمع بين الاثنين يمنع نجاحًا برمجيًا يخفي خطأً في تفسير المسألة.

There is also a distinction between testing and validation. A software test asks whether code satisfies its contract, while scientific validation asks whether the result answers the stated question appropriately. Combining both prevents a software success from hiding a scientific interpretation error.

عند ظهور نتيجة غير متوقعة، يجب أن يكون المسار التشخيصي مرتبًا: تحقق من المدخلات، ثم التحويلات، ثم الخوارزمية، ثم المقياس، ثم المصدر المرجعي. القفز مباشرة إلى تغيير النموذج يجعل من الصعب معرفة سبب التحسن أو التدهور.

When an unexpected result appears, diagnosis should follow an ordered path: inputs, transformations, algorithm, metric, and reference source. Jumping directly to a new model makes it difficult to know why performance improved or deteriorated.

وأخيرًا، ينبغي أن يبقى هذا القسم مرتبطًا ببقية المشروع. النتيجة ليست نهاية الخط؛ يمكن أن تصبح سجلًا في SQL، أو مدخلًا لتقرير، أو سببًا لطلب مراجعة، أو إشارة إلى أن المرحلة التالية غير مسموحة بعد. بهذه الطريقة يتحول المفهوم إلى جزء من نظام متكامل.

Finally, this section should remain connected to the rest of the project. A result is not necessarily the end of the pipeline; it can become a SQL record, a report input, a reason for review, or a signal that the next stage is not yet permitted. The concept thus becomes part of an integrated system.

فصل الاستدلال عن القرار

Separating Inference from Decision

في هذا القسم، نعامل طبقة القرار كجزء من تصميم النظام وليس كإضافة تجميلية. يجب أن يكون له تعريف واضح، ومدخلات قابلة للتتبع، ومخرج يمكن اختباره. إذا لم نستطع تحديد ما يدخل إلى العملية وما يخرج منها، فلن نستطيع تفسير نتيجة تغيرها بين تشغيل وآخر.

In this section, decision is treated as part of system design rather than as decoration. It needs a clear definition, traceable inputs, and testable outputs. If we cannot state what enters the process and what leaves it, we cannot explain why a result changes between runs.

الخطوة العملية هي تسجيل القرار إلى جانب النتيجة. لا يكفي أن نحتفظ بالرقم النهائي؛ نحتاج إلى معرفة الإصدار والبيانات والمعاملات التي أنتجته. هذا السجل يحمي المشروع من مشكلة إعادة الإنتاج التي تظهر عندما يعمل المثال مرة ثم يفشل بعد تحديث بيئة أو مصدر بيانات.

The practical step is to record the decision together with the result. Keeping only the final number is insufficient; we need the version,

data, and parameters that produced it. This protects the project from reproducibility failures that appear when an example works once and changes after an environment or data-source update.

هناك أيضًا فرق بين الاختبار والتحقق. الاختبار يسأل هل الكود يحقق عقده البرمجي، بينما التحقق العلمي يسأل هل الناتج يجيب عن السؤال المحدد بطريقة معقولة. الجمع بين الاثنين يمنع نجاحًا برمجيًا يخفي خطأً في تفسير المسألة.

There is also a distinction between testing and validation. A software test asks whether code satisfies its contract, while scientific validation asks whether the result answers the stated question appropriately. Combining both prevents a software success from hiding a scientific interpretation error.

عند ظهور نتيجة غير متوقعة، يجب أن يكون المسار التشخيصي مرتبًا: تحقق من المدخلات، ثم التحويلات، ثم الخوارزمية، ثم المقياس، ثم المصدر المرجعي. القفز مباشرة إلى تغيير النموذج يجعل من الصعب معرفة سبب التحسن أو التدهور.

When an unexpected result appears, diagnosis should follow an ordered path: inputs, transformations, algorithm, metric, and reference source. Jumping directly to a new model makes it difficult to know why performance improved or deteriorated.

وأخيرًا، ينبغي أن يبقى هذا القسم مرتبطًا ببقية المشروع. النتيجة ليست نهاية الخط؛ يمكن أن تصبح سجلًا في SQL، أو مدخلًا لتقرير، أو سببًا لطلب مراجعة، أو إشارة إلى أن المرحلة التالية غير مسموحة بعد. بهذه الطريقة يتحول المفهوم إلى جزء من نظام متكامل.

Finally, this section should remain connected to the rest of the project. A result is not necessarily the end of the pipeline; it can become a SQL record, a report input, a reason for review, or a signal that the next stage is not yet permitted. The concept thus becomes part of an integrated system.

شفافية مسؤولة

Responsible Transparency

في هذا القسم، نعامل الشفافية والخصوصية كجزء من تصميم النظام وليس كإضافة تجميلية. يجب أن يكون له تعريف واضح، ومدخلات قابلة للتتبع، ومخرج يمكن اختباره. إذا لم نستطع تحديد ما يدخل إلى العملية وما يخرج منها، فلن نستطيع تفسير نتيجة تغيرها بين تشغيل وآخر.

In this section, transparency is treated as part of system design rather than as decoration. It needs a clear definition, traceable inputs, and testable outputs. If we cannot state what enters the process and what leaves it, we cannot explain why a result changes between runs.

الخطوة العملية هي تسجيل القرار إلى جانب النتيجة. لا يكفي أن نحتفظ بالرقم النهائي؛ نحتاج إلى معرفة الإصدار والبيانات والمعاملات التي أنتجته. هذا السجل يحمي المشروع من مشكلة إعادة الإنتاج التي تظهر عندما يعمل المثال مرة ثم يفشل بعد تحديث بيئة أو مصدر بيانات.

The practical step is to record the decision together with the result. Keeping only the final number is insufficient; we need the version, data, and parameters that produced it. This protects the project from reproducibility failures that appear when an example works once and changes after an environment or data-source update.

هناك أيضًا فرق بين الاختبار والتحقق. الاختبار يسأل هل الكود يحقق عقده البرمجي، بينما التحقق العلمي يسأل هل الناتج يجيب عن السؤال المحدد بطريقة معقولة. الجمع بين الاثنين يمنع نجاحًا برمجيًا يخفي خطأً في تفسير المسألة.

There is also a distinction between testing and validation. A software test asks whether code satisfies its contract, while scientific validation asks whether the result answers the stated question appropriately. Combining both prevents a software success from hiding a scientific interpretation error.

عند ظهور نتيجة غير متوقعة، يجب أن يكون المسار التشخيصي مرتبًا: تحقق من المدخلات، ثم التحويلات، ثم الخوارزمية، ثم المقياس، ثم المصدر المرجعي. القفز مباشرة إلى تغيير النموذج يجعل من الصعب معرفة سبب التحسن أو التدهور.

When an unexpected result appears, diagnosis should follow an ordered path: inputs, transformations, algorithm, metric, and reference source. Jumping directly to a new model makes it difficult to know why performance improved or deteriorated.

وأخيرًا، ينبغي أن يبقى هذا القسم مرتبطًا ببقية المشروع. النتيجة ليست نهاية الخط؛ يمكن أن تصبح سجلًا في SQL، أو مدخلًا لتقرير، أو سببًا لطلب مراجعة، أو إشارة إلى أن المرحلة التالية غير مسموحة بعد. بهذه الطريقة يتحول المفهوم إلى جزء من نظام متكامل.

Finally, this section should remain connected to the rest of the project. A result is not necessarily the end of the pipeline; it can become a SQL record, a report input, a reason for review, or a signal that the next stage is not yet permitted. The concept thus becomes part of an integrated system.

الفصل السادس والعشرون: من قابلية التشغيل إلى قابلية المراجعة

Chapter 26: From Reproducibility to Auditability

26.1 عقد البيانات

26.1 The Data Contract

قبل أن تدخل البيانات إلى المصفوفة، يمكن تعريف عقد بيانات بسيط يحدد أسماء الأعمدة، وأنواعها، ونطاقاتها، والمفاتيح الفريدة، والقيم التي يسمح بها الحقل. هذا العقد يمنع اختلافًا صامتًا بين ملف المصدر والبرنامج. فإذا تغير اسم atomic_weight إلى weight مثلًا، يجب أن يفشل التحقق بوضوح بدل أن ينتج مصفوفة ناقصة ثم يواصل التحليل.

Before data enter the matrix, a simple data contract can define column names, types, allowed ranges, unique keys, and permitted values. This prevents silent drift between the source file and the program. If atomic_weight is renamed to weight, validation should

fail clearly instead of producing a partial matrix and continuing the analysis.

26.2 بصمة المخرجات

26.2 Output Fingerprints

للملف مع رقم الإصدار ووقت الإنشاء. البصمة لا تثبت hash عند إنشاء تقرير أو مصفوفة أو ملف نتائج، يمكن حفظ أن المحتوى صحيح علميًا، لكنها تثبت أن الملف الذي تمت مراجعته هو الملف الذي بقي بعد النقل أو النشر. هذه طبقة بسيطة لكنها قوية في المشاريع التي تنتقل فيها النتائج بين أجهزة أو أعضاء فريق مختلفين.

When a report, matrix, or result file is created, its hash can be stored with the version and creation time. A hash does not prove scientific correctness, but it helps prove that the reviewed file is the same file later transferred or published. This is a simple but useful layer when results move between machines or team members.

26.3 التكامل المستمر للاختبارات

26.3 Continuous Testing

حتى مشروع تعليمي يمكن أن يستفيد من تشغيل الاختبارات تلقائيًا عند تعديل الكود. لا نحتاج إلى بنية مؤسسية ضخمة؛ وسلامة واجهة الترميز ،NaN يكفي أن تتأكد العملية من عدد العناصر، وتفرد الرموز، وشكل المصفوفة، وغياب الكمومي. إذا فشل اختبار واحد، لا ينبغي نشر التقرير النهائي وكأنه سليم.

Even an educational project can benefit from automatically running tests after code changes. A large enterprise platform is not required; the process can simply check element count, symbol uniqueness, matrix shape, absence of NaN, and the quantum-encoding contract. If one test fails, the final report should not be published as if the system were healthy.

26.4 التقرير يذكر ما لم ينجح

26.4 The Report Records What Did Not Work

التقرير العلمي الجيد لا يعرض النتائج الناجحة فقط. إذا تعذر تشغيل جزء كمومي بسبب عدم توفر بيئة أو جهاز، يسجل التقرير ذلك بوضوح. وإذا كانت نتيجة نموذج غير مستقرة عبر البذور، يسجل عدم الاستقرار. هذه الشفافية تمنع القارئ من الخلط بين اكتمال الكتاب واكتمال كل تجربة في كل بيئة ممكنة.

A good scientific report does not show only successful results. If a quantum stage cannot run because an environment or device is unavailable, the report should say so. If a model is unstable across random seeds, that instability should be recorded. This prevents readers from confusing completeness of the book with successful execution in every possible environment.

26.5 SQL لاسترجاع أفضل النتائج

26.5 SQL for Best-Result Retrieval

يجب أن تكون الاستعلامات جزءًا من التصميم وليس ملحقًا بعديًا. يمكن استخراج quantum_runs، عند تخزين أفضل طاقة لكل نظام وخلفية، ثم مقارنة الإصدارات، ثم اختيار أحدث تشغيل. كما يمكن حساب عدد القيم الفريدة والانحراف المعياري عندما تكون عدة تشغيلات متاحة. هذه الاستعلامات تجعل الجدول سجلًا للتجارب لا مجرد مخزن أرقام.

When quantum_runs stores results, retrieval queries should be part of the design rather than an afterthought. The system can return the best energy per system and backend, compare software versions, and identify the latest run. It can also calculate the number of unique values and standard deviation when repeated runs exist. The table then becomes an experimental record rather than a numeric dump.

26.6 مقارنة إصدارات البرمجيات

26.6 Comparing Software Versions

إذا تغيرت نسخة Qiskit أو PySCF، فلا ينبغي دمج النتائج القديمة والجديدة في متوسط واحد دون تسجيل الإصدار. المقارنة بين الإصدارات تحتاج إلى تثبيت بقية الشروط قدر الإمكان، ثم تسجيل الفروق. هذا هو السبب في وجود software_version داخل سجل النتائج. الإصدار ليس معلومة إدارية؛ قد يغير السلوك العددي أو الواجهة أو خوارزمية التنفيذ.

When Qiskit or PySCF changes, old and new results should not be silently pooled into one average. Version comparisons require the other conditions to remain as similar as possible and the differences to be recorded. This is why software_version belongs in the result record. A version is not merely administrative metadata; it can affect numerical behavior, APIs, and execution details.

التقرير بين الإنسان والآلة 26.7

26.7 Reports Between Humans and Machines

قابلية القراءة جزء من السلامة 26.8

26.8 Readability Is Part of Safety

مثال مكتمل ومشروع إنتاجي 26.9

26.9 A Complete Example Versus a Production System

معيار الإغلاق النهائي 26.10

26.10 The Final Closure Criterio

اختبار قابلية الاستخراج 26.11

26.11 Testing Text Extractability

الفصل 27: هندسة المعاملات وترقية المخطط — من قاعدة تعمل إلى قاعدة يمكن تطويرها بأمان

Chapter 27: Transaction Engineering and Schema Migration — From a Working Database to a Safely Evolvable Database

هذا الفصل يضيف طبقة عملية إلى المشروع الحالي. الفكرة ليست زيادة عدد الأدوات، بل جعل المسار من البيانات إلى النتيجة أكثر وضوحًا وقابلية للقياس والمراجعة. كل مثال هنا يستخدم بيانات المشروع أو بيانات تعليمية مصطنعة بوضوح، ولا يعرض نتيجة حسابية على أنها حقيقة تجريبية لم تُختبر.

This chapter adds an operational layer to the existing project. The goal is not to increase the number of tools, but to make the path from data to result clearer, measurable, and reviewable. Each example uses project data or explicitly synthetic educational data, and no computational result is presented as experimental fact unless it has been independently tested.

27.1 لماذا لا يكفي إنشاء المخطط مرة واحدة

27.1 Why Creating the Schema Once Is Not Enough

في قاعدة البيانات ودورة حياتها لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «لماذا لا يكفي إنشاء المخطط مرة واحدة»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In database lifecycle management, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Why Creating the Schema Once Is Not Enough”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة

«لماذا لا يكفي إنشاء المخطط مرة واحدة» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Why Creating the Schema Once Is Not Enough” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

27.2 المعاملات والذرية في خط التحليل

27.2 Transactions and Atomicity in the Analysis Pipeline

في قاعدة البيانات ودورة حياتها لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «المعاملات والذرية في خط التحليل»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In database lifecycle management, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Transactions and Atomicity in the Analysis Pipeline”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «المعاملات والذرية في خط التحليل» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Transactions and Atomicity in the Analysis Pipeline” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

SQL

```sql
BEGIN IMMEDIATE;
ALTER TABLE elements ADD COLUMN data_version TEXT NOT NULL DEFAULT 'v1';
UPDATE elements SET data_version = 'v2';
COMMIT;
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests,

environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

ترقية المخطط دون فقد البيانات 27.3

27.3 Schema Migration Without Data Loss

في قاعدة البيانات ودورة حياتها لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «ترقية المخطط دون فقد البيانات»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In database lifecycle management, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Schema Migration Without Data Loss”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative

overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «ترقية المخطط دون فقد البيانات» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Schema Migration Without Data Loss” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

☐ هل حددت المدخلات والوحدات والإصدار قبل الحساب؟

☐ Were the inputs, units, and version defined before computation?

☐ هل يمكن إعادة بناء النتيجة من السجل نفسه؟

☐ Can the result be reconstructed from the same record?

☐ هل فصلت المثال التعليمي عن الدليل التجريبي؟

☐ Did you separate the educational example from experimental evidence?

27.4 البيانات الزمنية والإصدارات

27.4 Temporal Data and Versioning

في قاعدة البيانات ودورة حياتها لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «البيانات الزمنية والإصدارات»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In database lifecycle management, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Temporal Data and Versioning”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «البيانات الزمنية والإصدارات» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware.

Therefore, the result of “Temporal Data and Versioning” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
import sqlite3

with sqlite3.connect("data/periodic_table.db") as con:
    con.execute("PRAGMA foreign_keys = ON")
    con.execute("UPDATE elements SET data_version = ? WHERE atomic_number = ?", ("v2", 1))
    con.commit()
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

27.5 القيود كآلية حماية تشغيلية

27.5 Constraints as Operational Safety Controls

في قاعدة البيانات ودورة حياتها لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «القيود كآلية حماية تشغيلية»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In database lifecycle management, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Constraints as Operational Safety Controls”, this turns the workflow into an explicit chain of assumptions, data,

computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «القيود كآلية حماية تشغيلية» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Constraints as Operational Safety Controls” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

اختبارات الهجرة والرجوع 27.6

27.6 Migration and Rollback Tests

في قاعدة البيانات ودورة حياتها لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «اختبارات الهجرة والرجوع»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In database lifecycle management, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Migration and Rollback Tests”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة

«اختبارات الهجرة والرجوع» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Migration and Rollback Tests” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

SQL

```sql
BEGIN IMMEDIATE;
ALTER TABLE elements ADD COLUMN data_version TEXT NOT NULL DEFAULT 'v1';
UPDATE elements SET data_version = 'v2';
COMMIT;
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

هل حددت المدخلات والوحدات والإصدار قبل الحساب؟ ☐

☐ Were the inputs, units, and version defined before computation?

هل يمكن إعادة بناء النتيجة من السجل نفسه؟ ☐

☐ Can the result be reconstructed from the same record?

هل فصلت المثال التعليمي عن الدليل التجريبي؟ ☐

☐ Did you separate the educational example from experimental evidence?

مثال عملي على ترقية جدول العناصر 27.7

27.7 Practical Element-Table Migration Example

في قاعدة البيانات ودورة حياتها لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «مثال عملي على ترقية جدول العناصر»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In database lifecycle management, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Practical Element-Table Migration Example”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «مثال عملي على ترقية جدول العناصر» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Practical Element-Table Migration Example” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

ما الذي يجب تسجيله في سجل التغيير 27.8

27.8 What Belongs in a Change Log

في قاعدة البيانات ودورة حياتها لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «ما الذي يجب تسجيله في سجل التغيير»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In database lifecycle management, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “What Belongs in a Change Log”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو

دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «ما الذي يجب تسجيله في سجل التغيير» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “What Belongs in a Change Log” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
import sqlite3

with sqlite3.connect("data/periodic_table.db") as con:
    con.execute("PRAGMA foreign_keys = ON")
    con.execute("UPDATE elements SET data_version = ? WHERE atomic_number = ?", ("v2", 1))
    con.commit()
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

PYTHON

```python
from pathlib import Path
import hashlib

path = Path("artifacts/test.db")
digest = hashlib.sha256(path.read_bytes()).hexdigest()
print("migration_db_sha256", digest)
```

الترقية الآمنة تحتاج إلى نسخة اختبارية من قاعدة البيانات. يبدأ الاختبار بعدد سجلات معروف، ثم تُنفذ الترقية، ثم تُقارن أعداد الصفوف والمفاتيح والقيود قبل وبعد. يجب أيضًا اختبار أن التطبيق القديم لا يكتب في مخطط لم يعد يفهمه، وأن التطبيق الجديد يستطيع قراءة البيانات القديمة. يمكن حفظ بصمة ملف قاعدة البيانات ونسخة المخطط ونتيجة الاختبارات معًا، فيصبح تغيير المخطط حدثًا قابلًا للمراجعة بدل تعديل يدوي يختفي أثره.

A safe migration requires a test copy of the database. Start with a known row count, apply the migration, and compare rows, keys, and constraints before and after. Also test that the old application cannot silently write to a schema it no longer understands, while the new application can still read legacy data. Store the database fingerprint, schema version, and test results together so a schema change becomes an auditable event rather than a manual edit whose history disappears.

27.10 اختبار الترقية قبل اعتمادها

27.10 Testing a Migration Before Release

SQL

```sql
BEGIN TRANSACTION;
ALTER TABLE elements ADD COLUMN data_version TEXT;
UPDATE elements SET data_version = 'v2';
CREATE INDEX IF NOT EXISTS idx_elements_version
ON elements(data_version);
```

```
SELECT COUNT(*) AS missing_versions
FROM elements WHERE data_version IS NULL;
COMMIT;
```

في قاعدة البيانات العلمية، لا يكفي أن تكون الجداول صحيحة كلٌّ على حدة؛ يجب أن تكون عملية التغيير نفسها ذرية. إذا أضيف حقل جديد ثم فشلت الفهارس أو نقل البيانات، فلا ينبغي أن تبقى القاعدة في حالة نصف مُرقّاة. تبدأ الترقية بفحص الإصدار الحالي، ثم إنشاء البنية، ونقل البيانات، واختبار القيود، وتسجيل الإصدار الجديد داخل معاملة يمكن التراجع عنها. وعند ارتباط SQL بطبقة OOP يجب ألا يعلن الكائن نجاح الحفظ قبل نجاح المعاملة التي تحفظ حالته فعليًا.

Scientific databases need atomic change operations, not only correct individual tables. If a new field is added and an index or data migration then fails, the database should not remain half-migrated. A migration should inspect the current schema version, create the new structure, move the data, validate constraints, and record the new version inside a transaction that can be rolled back. When SQL is connected to an OOP layer, an object should not report a successful save before the transaction that actually persists its state has committed.

27.9 حدود المعاملة والترقية الذرية

27.9 Transaction Boundaries and Atomic Migrations

الفصل 28: هندسة الاستعلام والأداء — من صحة SQL إلى قابلية التوسع

Chapter 28: Query and Performance Engineering — From Correct SQL to Scalable SQL

هذا الفصل يضيف طبقة عملية إلى المشروع الحالي. الفكرة ليست زيادة عدد الأدوات، بل جعل المسار من البيانات إلى النتيجة أكثر وضوحًا وقابلية للقياس والمراجعة. كل مثال هنا يستخدم بيانات المشروع أو بيانات تعليمية مصطنعة بوضوح، ولا يعرض نتيجة حسابية على أنها حقيقة تجريبية لم تُختبر.

This chapter adds an operational layer to the existing project. The goal is not to increase the number of tools, but to make the path

from data to result clearer, measurable, and reviewable. Each example uses project data or explicitly synthetic educational data, and no computational result is presented as experimental fact unless it has been independently tested.

28.1 تكلفة الاستعلام جزء من تصميمه

28.1 Query Cost Is Part of Query Design

في الاستعلامات وخطط التنفيذ لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «تكلفة الاستعلام جزء من تصميمه»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In query and execution-plan engineering, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Query Cost Is Part of Query Design”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative

overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «تكلفة الاستعلام جزء من تصميمه» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Query Cost Is Part of Query Design” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

28.2 الفهارس ومقايضة القراءة والكتابة

28.2 Indexes and the Read-Write Trade-off

في الاستعلامات وخطط التنفيذ لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «الفهارس ومقايضة القراءة والكتابة»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In query and execution-plan engineering, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Indexes and the Read-Write Trade-off”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «الفهارس ومقايضة القراءة والكتابة» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Indexes and the Read-Write Trade-off” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

SQL

```
EXPLAIN QUERY PLAN
SELECT symbol, atomic_weight
FROM elements
WHERE group_number = ?
ORDER BY atomic_weight;
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

28.3 EXPLAIN وقراءة خطة التنفيذ

28.3 EXPLAIN and Reading the Query Plan

في الاستعلامات وخطط التنفيذ لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. وقراءة خطة التنفيذ»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات EXPLAIN» عند تطبيق ذلك على ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In query and execution-plan engineering, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “EXPLAIN and Reading the Query Plan”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column

order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «EXPLAIN وقراءة خطة التنفيذ» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “EXPLAIN and Reading the Query Plan” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

هل حددت المدخلات والوحدات والإصدار قبل الحساب؟ ☐

☐ Were the inputs, units, and version defined before computation?

هل يمكن إعادة بناء النتيجة من السجل نفسه؟ ☐

☐ Can the result be reconstructed from the same record?

هل فصلت المثال التعليمي عن الدليل التجريبي؟ ☐

☐ Did you separate the educational example from experimental evidence?

التجميع والفرز والذاكرة 28.4

28.4 Aggregation, Sorting, and Memory

في الاستعلامات وخطط التنفيذ لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «التجميع والفرز والذاكرة»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In query and execution-plan engineering, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Aggregation, Sorting, and Memory”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة

«التجميع والفرز والذاكرة» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Aggregation, Sorting, and Memory” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```python
import sqlite3

with sqlite3.connect("data/periodic_table.db") as con:
    rows = con.execute(
        "SELECT symbol, atomic_weight FROM elements WHERE group_number = ? ORDER BY
atomic_weight",
        (17,),
    ).fetchall()
print(rows)
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

28.5 الاستعلامات المعلّمة وحماية طبقة البيانات

28.5 Parameterized Queries and Data-Layer Safety

في الاستعلامات وخطط التنفيذ لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه.

عند تطبيق ذلك على «الاستعلامات المعلّمة وحماية طبقة البيانات»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In query and execution-plan engineering, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Parameterized Queries and Data-Layer Safety”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «الاستعلامات المعلّمة وحماية طبقة البيانات» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an

ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Parameterized Queries and Data-Layer Safety” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

28.6 النتائج المادية والتخزين المؤقت

28.6 Materialized Results and Caching

في الاستعلامات وخطط التنفيذ لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «النتائج المادية والتخزين المؤقت»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In query and execution-plan engineering, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Materialized Results and Caching”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software

versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «النتائج المادية والتخزين المؤقت» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Materialized Results and Caching” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

SQL

```
EXPLAIN QUERY PLAN
SELECT symbol, atomic_weight
FROM elements
WHERE group_number = ?
ORDER BY atomic_weight;
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

هل حددت المدخلات والوحدات والإصدار قبل الحساب؟ ☐

☐ Were the inputs, units, and version defined before computation?

هل يمكن إعادة بناء النتيجة من السجل نفسه؟ ☐

☐ Can the result be reconstructed from the same record?

هل فصلت المثال التعليمي عن الدليل التجريبي؟ ☐

☐ Did you separate the educational example from experimental evidence?

28.7 قياس الأداء بدل تخمينه

28.7 Measure Performance Instead of Guessing

في الاستعلامات وخطط التنفيذ لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «قياس الأداء بدل تخمينه»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In query and execution-plan engineering, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Measure Performance Instead of Guessing”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be

defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «قياس الأداء بدل تخمينه» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Measure Performance Instead of Guessing” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

28.8 متى ننتقل من SQLite إلى محرك أكبر

28.8 When to Move Beyond SQLite

في الاستعلامات وخطط التنفيذ لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «متى ننتقل من SQLite إلى محرك أكبر»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In query and execution-plan engineering, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “When to Move Beyond SQLite”, this turns the workflow

into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «متى ننتقل من SQLite إلى محرك أكبر» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “When to Move Beyond SQLite” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
import sqlite3

with sqlite3.connect("data/periodic_table.db") as con:
    rows = con.execute(
        "SELECT symbol, atomic_weight FROM elements WHERE group_number = ? ORDER BY 
atomic_weight",
        (17,),
    ).fetchall()
print(rows)
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

PYTHON / benchmark_sql.py

```
from __future__ import annotations
import sqlite3
import time

DB = "data/periodic_table.db"
QUERY = "SELECT category_id, COUNT(*) FROM elements GROUP BY category_id"

with sqlite3.connect(DB) as con:
    con.execute("PRAGMA foreign_keys = ON")
    start = time.perf_counter()
    rows = con.execute(QUERY).fetchall()
    elapsed_ms = (time.perf_counter() - start) * 1000

print({"query": QUERY, "elapsed_ms": round(elapsed_ms, 3), "row_count": len(rows)})
```

PYTHON

```
import sqlite3

con = sqlite3.connect("elements.db")
rows = con.execute(
    "SELECT symbol, atomic_weight FROM elements "
    "WHERE atomic_number = ?",
    (26,),
).fetchall()
print(rows)
con.close()
```

حتى في مشروع صغير يجب ألا تُبنى جمل SQL بضم السلاسل النصية من مدخل المستخدم.
الاستعلامات المعلّمة تفصل بين بنية SQL والقيمة، وتمنع خلط البيانات بالكود، وتجعل السلوك قابلاً

للاختبار. عند إضافة واجهة ويب أو خدمة، يمكن تسجيل زمن الاستعلام وعدد الصفوف وإصدار المخطط من دون تسجيل أسرار أو بيانات شخصية. الأمن هنا جزء من عقد الاستعلام نفسه.

Even in a small project, SQL statements should not be constructed by concatenating user input. Parameterized queries separate SQL structure from values, prevent data from being confused with code, and make behavior testable. When a web interface or service is added, query duration, returned-row count, and schema version can be logged without exposing secrets or personal data. Security is part of the query contract itself.

28.10 الاستعلامات المعلّمة وسلامة المدخلات

28.10 Parameterized Queries and Input Safety

SQL

```
EXPLAIN QUERY PLAN
SELECT symbol, atomic_weight
FROM elements
WHERE atomic_number BETWEEN 20 AND 80
ORDER BY atomic_number;

CREATE INDEX IF NOT EXISTS idx_elements_atomic_number
ON elements(atomic_number);

EXPLAIN QUERY PLAN
SELECT symbol, atomic_weight
FROM elements
WHERE atomic_number BETWEEN 20 AND 80
ORDER BY atomic_number;
```

إضافة الفهرس قرار قابل للقياس. يبدأ التحسين من السؤال: ما الاستعلام البطيء، وكم صفًا يقرأ، وما خطة التنفيذ التي اختارها المحرك؟ في SQLite يمكن استخدام EXPLAIN QUERY PLAN لمعرفة ما إذا كان الاستعلام يستفيد من فهرس أو ما زال ينفذ مسحًا كاملًا. إذا لم تتغير الخطة أو أصبحت عمليات الكتابة أثقل من الفائدة، فلا يوجد سبب هندسي لإبقاء الفهرس. هكذا يتحول الأداء من انطباع إلى تجربة قابلة لإعادة الإنتاج.

An index is a measurable optimization. Start by asking which query is slow, how many rows it reads, and what execution plan the engine selected. SQLite's EXPLAIN QUERY PLAN can show whether a query uses an index or still performs a full table scan. If the plan does not

improve, or write costs become larger than the benefit, there is no engineering reason to keep the index. Performance becomes a reproducible experiment rather than an impression.

28.9 خطة التنفيذ قبل إضافة فهرس

28.9 Inspect the Query Plan Before Adding an Index

الفصل 29: جودة البيانات العددية — القيم المفقودة والوحدات وعدم اليقين

Chapter 29: Numerical Data Quality — Missing Values, Units, and Uncertainty

هذا الفصل يضيف طبقة عملية إلى المشروع الحالي. الفكرة ليست زيادة عدد الأدوات، بل جعل المسار من البيانات إلى النتيجة أكثر وضوحًا وقابلية للقياس والمراجعة. كل مثال هنا يستخدم بيانات المشروع أو بيانات تعليمية مصطنعة بوضوح، ولا يعرض نتيجة حسابية على أنها حقيقة تجريبية لم تُختبر.

This chapter adds an operational layer to the existing project. The goal is not to increase the number of tools, but to make the path from data to result clearer, measurable, and reviewable. Each example uses project data or explicitly synthetic educational data, and no computational result is presented as experimental fact unless it has been independently tested.

29.1 البيانات العددية ليست أرقامًا فقط

29.1 Numerical Data Are Not Just Numbers

في القياس العددي وجودة البيانات لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «البيانات العددية ليست أرقامًا فقط»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In numerical measurement and data quality, the goal should not be to add a technical layer merely because it exists. The first questions

are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Numerical Data Are Not Just Numbers”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «البيانات العددية ليست أرقامًا فقط» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Numerical Data Are Not Just Numbers” must be stated with its limits: what was actually computed, what is an

educational example, and what requires independent data or experiment before it can be treated as evidence.

القيم المفقودة وآلية فقدها 29.2

29.2 Missing Values and Their Mechanisms

في القياس العددي وجودة البيانات لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «القيم المفقودة وآلية فقدها»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In numerical measurement and data quality, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Missing Values and Their Mechanisms”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «القيم المفقودة وآلية فقدها» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Missing Values and Their Mechanisms” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
import numpy as np

values = np.asarray([1.2, 1.3, np.nan, 1.1], dtype=float)
valid = np.isfinite(values)
mean = values[valid].mean()
print({"n": int(valid.sum()), "mean": float(mean)})
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

29.3 الوحدات والتحويلات

29.3 Units and Conversions

في القياس العددي وجودة البيانات لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه.

عند تطبيق ذلك على «الوحدات والتحويلات»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In numerical measurement and data quality, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Units and Conversions”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «الوحدات والتحويلات» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an

ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Units and Conversions” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

هل حددت المدخلات والوحدات والإصدار قبل الحساب؟ ☐

☐ Were the inputs, units, and version defined before computation?

هل يمكن إعادة بناء النتيجة من السجل نفسه؟ ☐

☐ Can the result be reconstructed from the same record?

هل فصلت المثال التعليمي عن الدليل التجريبي؟ ☐

☐ Did you separate the educational example from experimental evidence?

29.4 عدم اليقين وفواصل الثقة

29.4 Uncertainty and Confidence Intervals

في القياس العددي وجودة البيانات لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «عدم اليقين وفواصل الثقة»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In numerical measurement and data quality, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Uncertainty and Confidence Intervals”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «عدم اليقين وفواصل الثقة» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Uncertainty and Confidence Intervals” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```python
from dataclasses import dataclass

@dataclass(frozen=True)
class Measurement:
    value: float
    unit: str
    uncertainty: float | None
```

```
m = Measurement(1.008, "u", 0.001)
print(m)
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

29.5 القيم الشاذة والتشخيص

29.5 Outliers and Diagnostics

في القياس العددي وجودة البيانات لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «القيم الشاذة والتشخيص»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In numerical measurement and data quality, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Outliers and Diagnostics”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «القيم الشاذة والتشخيص» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Outliers and Diagnostics” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

29.6 الاستقرار العددي والتحقق من المدخلات

29.6 Numerical Stability and Input Validation

في القياس العددي وجودة البيانات لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «الاستقرار العددي والتحقق من المدخلات»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In numerical measurement and data quality, the goal should not be to add a technical layer merely because it exists. The first questions

are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Numerical Stability and Input Validation”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «الاستقرار العددي والتحقق من المدخلات» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Numerical Stability and Input Validation” must be stated with its limits: what was actually computed, what is

an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
import numpy as np

values = np.asarray([1.2, 1.3, np.nan, 1.1], dtype=float)
valid = np.isfinite(values)
mean = values[valid].mean()
print({"n": int(valid.sum()), "mean": float(mean)})
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

☐ هل حددت المدخلات والوحدات والإصدار قبل الحساب؟

☐ Were the inputs, units, and version defined before computation?

☐ هل يمكن إعادة بناء النتيجة من السجل نفسه؟

☐ Can the result be reconstructed from the same record?

☐ هل فصلت المثال التعليمي عن الدليل التجريبي؟

☐ Did you separate the educational example from experimental evidence?

29.7 المصفوفات المتناثرة والذاكرة

29.7 Sparse Matrices and Memory

في القياس العددي وجودة البيانات لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «المصفوفات المتناثرة والذاكرة»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In numerical measurement and data quality, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Sparse Matrices and Memory”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «المصفوفات المتناثرة والذاكرة» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware.

Therefore, the result of “Sparse Matrices and Memory” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

29.8 عقد البيانات بين SQL وNumPy

29.8 Data Contracts Between SQL and NumPy

في القياس العددي وجودة البيانات لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها عند تطبيق ذلك على «عقد البيانات بين SQL وNumPy».

In numerical measurement and data quality, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Data Contracts Between SQL and NumPy”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative

overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «عقد البيانات بين SQL وNumPy» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Data Contracts Between SQL and NumPy” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
from dataclasses import dataclass

@dataclass(frozen=True)
class Measurement:
    value: float
    unit: str
    uncertainty: float | None

m = Measurement(1.008, "u", 0.001)
print(m)
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

PYTHON

```
import numpy as np

values = np.array([1.0, np.nan, 3.0, np.nan])
missing = np.isnan(values)
mean_train = np.nanmean(values)
filled = np.where(missing, mean_train, values)
print("missing:", missing.tolist())
print("filled:", filled.tolist())
```

القيمة المفقودة ليست صفرًا، وعدم وجود القياس ليس دليلًا على عدم وجود الخاصية. يحتاج مسار البيانات إلى تمييز القيمة الموجودة والمفقودة وغير القابلة للتطبيق، وقد يحتاج إلى تمثيل عدم اليقين أيضًا. عند تدريب نموذج، يجب ألا تستخدم عملية التعويض معلومات من مجموعة الاختبار. كما ينبغي تسجيل قاعدة التعويض، لأن تغييرها قد يغير النموذج والنتيجة.

A missing value is not zero, and the absence of a measurement is not evidence that a property is absent. The data pipeline should distinguish observed, missing, and not-applicable states and may also need uncertainty information. During training, imputation must not use information from the test set. The imputation rule should be recorded because changing it can change both the model and its reported result.

29.10 تحليل القيم المفقودة وعدم اليقين

29.10 Missingness and Measurement Uncertainty

PYTHON

```
from dataclasses import dataclass

@dataclass(frozen=True)
class Measurement:
    value: float
    unit: str
    source: str

m = Measurement(273.15, "K", "reference")
print(m)
```

القيمة العددية بلا وحدة قد تكون صحيحة حسابيًا وخاطئة علميًا. يجب أن يحمل كل قياس وحدته ونطاقه ومصدره، وأن يكون التحويل بين الوحدات عملية واضحة ومسجلة. قد تظهر خصائص العناصر بوحدات مختلفة في مصادر مختلفة؛ وقبل إدخالها إلى مصفوفة التعلم الآلي يجب توحيد الوحدات مع الاحتفاظ بالوحدة الأصلية. إذا تغيرت وحدة عمود، فهذا تغيير في تعريف الميزة وليس مجرد تنسيق.

A number without a unit can be computationally valid and scientifically wrong. Every measurement should carry its unit, range, and source, and unit conversion should be an explicit recorded operation. Element properties may arrive in different units from different sources; before entering a machine-learning matrix, units should be normalized while the original unit is preserved. Changing a feature's unit changes its data definition rather than merely its formatting.

29.9 الوحدات ليست مجرد نصوص

29.9 Units Are Data, Not Just Labels

الفصل 30: المعالجة على نطاق البيانات — التجزئة والتدفق و Dask و Spark

Chapter 30: Data-Scale Engineering — Partitioning, Streaming, Dask, and Spark

هذا الفصل يضيف طبقة عملية إلى المشروع الحالي. الفكرة ليست زيادة عدد الأدوات، بل جعل المسار من البيانات إلى النتيجة أكثر وضوحًا وقابلية للقياس والمراجعة. كل مثال هنا يستخدم بيانات المشروع أو بيانات تعليمية مصطنعة بوضوح، ولا يعرض نتيجة حسابية على أنها حقيقة تجريبية لم تُختبر.

This chapter adds an operational layer to the existing project. The goal is not to increase the number of tools, but to make the path from data to result clearer, measurable, and reviewable. Each example uses project data or explicitly synthetic educational data, and no computational result is presented as experimental fact unless it has been independently tested.

30.1 لماذا لا نوزع 118 سجلًا؟

30.1 Why We Do Not Distribute 118 Records

في المعالجة الموزعة لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «لماذا لا نوزع 118 سجلًا؟»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In distributed data processing, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Why We Do Not Distribute 118 Records”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة

«لماذا لا نوزع 118 سجلًا؟» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of "Why We Do Not Distribute 118 Records" must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

30.2 من دفعة واحدة إلى دفعات قابلة للقياس

30.2 From One Batch to Measurable Batches

في المعالجة الموزعة لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «من دفعة واحدة إلى دفعات قابلة للقياس»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In distributed data processing, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to "From One Batch to Measurable Batches", this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «من دفعة واحدة إلى دفعات قابلة للقياس» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “From One Batch to Measurable Batches” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
import dask.dataframe as dd

df = dd.read_parquet("data/large_table/*.parquet")
result = (df[df["group_number"] == 17]
          .groupby("period")["atomic_weight"]
          .mean()
          .compute())
print(result)
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

30.3 Dask والذاكرة الأكبر من RAM

30.3 Dask and Data Larger Than RAM

في المعالجة الموزعة لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «Dask والذاكرة الأكبر من RAM»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In distributed data processing, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Dask and Data Larger Than RAM”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software

versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «Dask والذاكرة الأكبر من RAM» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Dask and Data Larger Than RAM” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

☐ هل حددت المدخلات والوحدات والإصدار قبل الحساب؟

☐ Were the inputs, units, and version defined before computation?

☐ هل يمكن إعادة بناء النتيجة من السجل نفسه؟

☐ Can the result be reconstructed from the same record?

☐ هل فصلت المثال التعليمي عن الدليل التجريبي؟

☐ Did you separate the educational example from experimental evidence?

30.4 Spark عندما تصبح البيانات موزعة فعلاً

30.4 Spark When Data Is Truly Distributed

في المعالجة الموزعة لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق

عندما تصبح البيانات موزعة فعلاً»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات Spark» ذلك على ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In distributed data processing, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Spark When Data Is Truly Distributed”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «Spark عندما تصبح البيانات موزعة فعلاً» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an

ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Spark When Data Is Truly Distributed” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("PeriodicTableScale").getOrCreate()
df = spark.read.parquet("data/large_table/")
df.groupBy("period").avg("atomic_weight").show()
spark.stop()
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

30.5 الاتساق بين التقسيم والنتيجة

30.5 Consistency Between Partitioning and Results

في المعالجة الموزعة لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «الاتساق بين التقسيم والنتيجة»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In distributed data processing, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Consistency Between Partitioning and Results”, this

turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «الاتساق بين التقسيم والنتيجة» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Consistency Between Partitioning and Results” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

نقاط الفشل وإعادة المحاولة 30.6

30.6 Failure Points and Retries

في المعالجة الموزعة لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «نقاط الفشل وإعادة المحاولة»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In distributed data processing, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Failure Points and Retries”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة

«نقاط الفشل وإعادة المحاولة» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Failure Points and Retries” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
import dask.dataframe as dd

df = dd.read_parquet("data/large_table/*.parquet")
result = (df[df["group_number"] == 17]
          .groupby("period")["atomic_weight"]
          .mean()
          .compute())
print(result)
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

☐ هل حددت المدخلات والوحدات والإصدار قبل الحساب؟

☐ Were the inputs, units, and version defined before computation?

☐ هل يمكن إعادة بناء النتيجة من السجل نفسه؟

☐ Can the result be reconstructed from the same record?

☐ هل فصلت المثال التعليمي عن الدليل التجريبي؟

☐ Did you separate the educational example from experimental evidence?

30.7 مراقبة الزمن والذاكرة والتكلفة

30.7 Monitoring Time, Memory, and Cost

في المعالجة الموزعة لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «مراقبة الزمن والذاكرة والتكلفة»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In distributed data processing, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Monitoring Time, Memory, and Cost”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «مراقبة الزمن والذاكرة والتكلفة» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Monitoring Time, Memory, and Cost” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

30.8 قرار اختيار المحرك

30.8 Choosing the Processing Engine

في المعالجة الموزعة لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «قرار اختيار المحرك»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In distributed data processing, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Choosing the Processing Engine”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو

دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «قرار اختيار المحرك» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Choosing the Processing Engine” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("PeriodicTableScale").getOrCreate()
df = spark.read.parquet("data/large_table/")
df.groupBy("period").avg("atomic_weight").show()
spark.stop()
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

PYTHON

```
import pandas as pd

sample = pd.DataFrame({
    "element": [1, 1, 2, 2],
    "value": [0.2, 0.4, 0.3, 0.5],
})
print(sample.groupby("element")["value"].mean())
```

Spark يختلف عن Dask في منظومته التشغيلية وقدرته على الاندماج مع بيئات موزعة واسعة. لا ينبغي إدخاله لأن اسمه أكبر، بل عندما يصبح حجم البيانات أو عدد المستخدمين أو متطلبات التكامل مع التخزين العنقودي مبررًا لذلك. يمكن مقارنة Dask وSpark على مهمة اصطناعية واحدة باستخدام زمن القراءة والحساب والذاكرة ونقل البيانات. المقارنة العادلة تثبت أن اختيار الأداة قرار هندسي مرتبط بحجم العمل.

Spark differs from Dask in its operational ecosystem and its fit for larger distributed environments. It should not be introduced because the name sounds bigger; it becomes reasonable when data volume, concurrency, or cluster integration justifies it. Compare Dask and Spark on the same synthetic aggregation using read time, compute time, memory behavior, and data transfer. A fair comparison shows that tool choice is an engineering decision tied to workload.

30.10 متى يصبح Spark منطقيًا؟

30.10 When Does Spark Become the Right Choice?

PYTHON

```
import dask.dataframe as dd

reads = dd.read_csv("data/reads-*.csv", blocksize="64MB")
```

```
print("partitions:", reads.npartitions)
print(reads.map_partitions(len).compute())
```

Dask لا يجعل مجموعة صغيرة كبيرة بطريقة سحرية؛ فائدته تظهر عندما يمكن تقسيم العمل إلى أجزاء مستقلة بكلفة مقبولة. حجم الجزء يؤثر في الذاكرة وعدد المهام وكلفة الجدولة. الأجزاء الصغيرة جدًا تزيد كلفة إدارة المهام، والكبيرة جدًا قد تتجاوز ذاكرة العامل. لذلك يجب قياس حجم الملفات وعدد الصفوف والذاكرة المتاحة، ثم مقارنة أكثر من تقسيم. في مثال جينومي اصطناعي يمكن أن يمثل كل ملف أو مجموعة سجلات جزءًا مع الاحتفاظ بهوية العينة وإصدار البيانات.

Dask does not make a small dataset large by magic; its value appears when work can be partitioned into independent pieces at an acceptable scheduling cost. Partition size affects memory, task count, and scheduling overhead. Very small partitions create excessive task management, while very large partitions can exceed worker memory. Measure file size, row count, and available memory, then compare partitioning strategies. In a synthetic genomic example, each file or record group can form a partition while sample identity and dataset version remain attached.

30.9 اختيار حجم الجزء في Dask

30.9 Choosing a Partition Size in Dask

الفصل 31: هندسة التعلم الآلي — منع التسرب والتحقق والمعايرة

Chapter 31: Machine-Learning Engineering — Leakage Prevention, Validation, and Calibration

هذا الفصل يضيف طبقة عملية إلى المشروع الحالي. الفكرة ليست زيادة عدد الأدوات، بل جعل المسار من البيانات إلى النتيجة أكثر وضوحًا وقابلية للقياس والمراجعة. كل مثال هنا يستخدم بيانات المشروع أو بيانات تعليمية مصطنعة بوضوح، ولا يعرض نتيجة حسابية على أنها حقيقة تجريبية لم تُختبر.

This chapter adds an operational layer to the existing project. The goal is not to increase the number of tools, but to make the path

from data to result clearer, measurable, and reviewable. Each example uses project data or explicitly synthetic educational data, and no computational result is presented as experimental fact unless it has been independently tested.

31.1 لماذا الدقة وحدها لا تكفي

31.1 Why Accuracy Alone Is Not Enough

في التعلم الآلي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «لماذا الدقة وحدها لا تكفي»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In machine learning, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Why Accuracy Alone Is Not Enough”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative

overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «لماذا الدقة وحدها لا تكفي» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Why Accuracy Alone Is Not Enough” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

31.2 تسرب البيانات قبل التدريب

31.2 Data Leakage Before Training

في التعلم الآلي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «تسرب البيانات قبل التدريب»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In machine learning, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Data Leakage Before Training”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «تسرب البيانات قبل التدريب» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Data Leakage Before Training” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
from sklearn.model_selection import GroupKFold, cross_val_score
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(random_state=7)
cv = GroupKFold(n_splits=5)
scores = cross_val_score(model, X, y, groups=experiment_id, cv=cv)
print(scores.mean())
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

31.3 تقسيم البيانات بطريقة تحترم بنية المسألة

31.3 Splitting Data According to the Problem Structure

في التعلم الآلي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «تقسيم البيانات بطريقة تحترم بنية المسألة»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In machine learning, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Splitting Data According to the Problem Structure”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column

order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «تقسيم البيانات بطريقة تحترم بنية المسألة» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Splitting Data According to the Problem Structure” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

☐ هل حددت المدخلات والوحدات والإصدار قبل الحساب؟

☐ Were the inputs, units, and version defined before computation?

☐ هل يمكن إعادة بناء النتيجة من السجل نفسه؟

☐ Can the result be reconstructed from the same record?

☐ هل فصلت المثال التعليمي عن الدليل التجريبي؟

☐ Did you separate the educational example from experimental evidence?

عدم توازن الفئات 31.4

31.4 Class Imbalance

في التعلم الآلي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «عدم توازن الفئات»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In machine learning, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Class Imbalance”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «عدم توازن الفئات» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Class Imbalance” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
import torch
import random
import numpy as np

seed = 7
random.seed(seed)
np.random.seed(seed)
torch.manual_seed(seed)
torch.use_deterministic_algorithms(True)
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

31.5 المعايرة والثقة الاحتمالية

31.5 Calibration and Probabilistic Confidence

في التعلم الآلي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «المعايرة والثقة الاحتمالية»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In machine learning, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Calibration and Probabilistic Confidence”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «المعايرة والثقة الاحتمالية» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware.

Therefore, the result of “Calibration and Probabilistic Confidence” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

الانجراف في البيانات 31.6

31.6 Data Drift

في التعلم الآلي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «الانجراف في البيانات»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In machine learning, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Data Drift”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative

overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «الانجراف في البيانات» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Data Drift” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
from sklearn.model_selection import GroupKFold, cross_val_score
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(random_state=7)
cv = GroupKFold(n_splits=5)
scores = cross_val_score(model, X, y, groups=experiment_id, cv=cv)
print(scores.mean())
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

☐ هل حددت المدخلات والوحدات والإصدار قبل الحساب؟

☐ Were the inputs, units, and version defined before computation?

☐ هل يمكن إعادة بناء النتيجة من السجل نفسه؟

☐ Can the result be reconstructed from the same record?

☐ هل فصلت المثال التعليمي عن الدليل التجريبي؟

☐ Did you separate the educational example from experimental evidence?

31.7 إعادة الإنتاج في PyTorch

31.7 Reproducibility in PyTorch

في التعلم الآلي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «إعادة الإنتاج في PyTorch»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In machine learning, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Reproducibility in PyTorch”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative

overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «إعادة الإنتاج في PyTorch» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Reproducibility in PyTorch” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

31.8 الاختبار بعد النشر

31.8 Post-Deployment Testing

في التعلم الآلي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «الاختبار بعد النشر»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In machine learning, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Post-Deployment Testing”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «الاختبار بعد النشر» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Post-Deployment Testing” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
import torch
import random
import numpy as np

seed = 7
random.seed(seed)
np.random.seed(seed)
```

```
torch.manual_seed(seed)
torch.use_deterministic_algorithms(True)
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

PYTHON / calibration_check.py

```
from __future__ import annotations
from sklearn.calibration import CalibratedClassifierCV
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import brier_score_loss, log_loss
import numpy as np

rng = np.random.default_rng(7)
X = rng.normal(size=(300, 4))
y = (X[:, 0] + 0.4 * X[:, 1] > 0).astype(int)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=7, stratify=y)
base = LogisticRegression(max_iter=2000).fit(X_train, y_train)
calibrated = CalibratedClassifierCV(base, method="sigmoid", cv=5).fit(X_train, y_train)
for name, model in [("base", base), ("calibrated", calibrated)]:
    p = model.predict_proba(X_test)[:, 1]
    print(name, "brier=", brier_score_loss(y_test, p), "logloss=", log_loss(y_test, p))
```

PYTHON

```
from sklearn.calibration import CalibratedClassifierCV
from sklearn.linear_model import LogisticRegression

base = LogisticRegression(max_iter=2000)
calibrated = CalibratedClassifierCV(
    base, method="sigmoid", cv=5
)
print(calibrated)
```

قد ينجح نموذج في التصنيف لكنه يعطي احتمالات مبالغًا فيها. لذلك لا يكفي رقم الدقة وحده؛ نحتاج إلى منحنى معايرة ومقاييس مثل Brier score أو log loss، مع فصل بيانات المعايرة عن الاختبار النهائي. في هذا المشروع لا تُستخدم احتمالات النموذج كحكم طبي؛ هي مخرجات عددية تحتاج إلى تفسير ومراجعة مستقلة. هذه النقطة مهمة خصوصًا عندما تُعرض النتائج في واجهة للمستخدم.

A classifier can achieve good accuracy while producing poorly calibrated probabilities. Accuracy alone is therefore insufficient; calibration curves and metrics such as Brier score or log loss provide additional evidence, with calibration data separated from the final test set. The project does not treat model probabilities as medical judgments; they remain numerical outputs requiring interpretation and independent review. This matters especially when results are displayed in a user interface.

31.10 المعايرة أهم من رقم الدقة وحده

31.10 Calibration Matters Beyond Accuracy

PYTHON

```
from sklearn.pipeline import make_pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression

model = make_pipeline(
    SimpleImputer(strategy="median"),
    StandardScaler(),
    LogisticRegression(max_iter=2000),
)
print(model)
```

تسرب البيانات يحدث عندما تدخل إلى التدريب معلومة لا تكون متاحة وقت التنبؤ الحقيقي. قد يحدث لأن عمودًا مشتقًّا من الهدف بقي في جدول الميزات، أو لأن التطبيع والتعويض تمّا على كامل البيانات قبل التقسيم. الحل هو بناء Pipeline يجعل التحويلات جزءًا من التدريب داخل كل طي من طيات التحقق. كما يجب تسجيل إصدار البيانات وإصدار بناء الميزات حتى يمكن تفسير تغير النتيجة.

Data leakage occurs when training receives information that would not be available at prediction time. A target-derived column may remain in the feature table, or preprocessing may be fitted on the full dataset before splitting. The solution is to build a Pipeline so transformations are fitted inside each validation split. Record the dataset version and feature-build version so changes in performance can be traced to a defined transformation.

31.9 منع التسرب بين الميزات والهدف

31.9 Preventing Feature Leakage

الفصل 32: الذكاء الاصطناعي التوليدي — من الاقتراح إلى الدليل

Chapter 32: Generative AI — From Proposal to Evidence

هذا الفصل يضيف طبقة عملية إلى المشروع الحالي. الفكرة ليست زيادة عدد الأدوات، بل جعل المسار من البيانات إلى النتيجة أكثر وضوحًا وقابلية للقياس والمراجعة. كل مثال هنا يستخدم بيانات المشروع أو بيانات تعليمية مصطنعة بوضوح، ولا يعرض نتيجة حسابية على أنها حقيقة تجريبية لم تُختبر.

This chapter adds an operational layer to the existing project. The goal is not to increase the number of tools, but to make the path from data to result clearer, measurable, and reviewable. Each example uses project data or explicitly synthetic educational data, and no computational result is presented as experimental fact unless it has been independently tested.

32.1 GAN وVAE: ما الذي تولده النماذج فعليًا؟

32.1 GANs and VAEs: What Do the Models Actually Generate?

في التوليد والاقتراح العلمي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «GAN وVAE: ما الذي تولده النماذج فعليًا؟»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific generation and proposal workflows, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “GANs and VAEs: What Do the Models Actually Generate?”, this turns the workflow into an explicit

chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «GAN وVAE: ما الذي تولده النماذج فعليًا؟» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “GANs and VAEs: What Do the Models Actually Generate?” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

لماذا 118 عنصرًا ليست مجموعة تدريب كافية لنموذج ضخم 32.2

32.2 Why 118 Elements Are Not a Sufficient Training Set for a Large Model

في التوليد والاقتراح العلمي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «لماذا 118 عنصرًا ليست مجموعة تدريب كافية لنموذج ضخم»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific generation and proposal workflows, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Why 118 Elements Are Not a Sufficient Training Set for a Large Model”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة

«لماذا 118 عنصرًا ليست مجموعة تدريب كافية لنموذج ضخم» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Why 118 Elements Are Not a Sufficient Training Set for a Large Model” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
from sklearn.neighbors import NearestNeighbors
from sklearn.preprocessing import StandardScaler

Xz = StandardScaler().fit_transform(X)
model = NearestNeighbors(n_neighbors=5).fit(Xz)
distance, index = model.kneighbors(target.reshape(1, -1))
print(index[0], distance[0])
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

32.3 التمثيل الجزيئي قبل التوليد

32.3 Molecular Representation Before Generation

في التوليد والاقتراح العلمي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند

تطبيق ذلك على «التمثيل الجزيئي قبل التوليد»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific generation and proposal workflows, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Molecular Representation Before Generation”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «التمثيل الجزيئي قبل التوليد» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an

ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Molecular Representation Before Generation” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

هل حددت المدخلات والوحدات والإصدار قبل الحساب؟ ☐

☐ Were the inputs, units, and version defined before computation?

هل يمكن إعادة بناء النتيجة من السجل نفسه؟ ☐

☐ Can the result be reconstructed from the same record?

هل فصلت المثال التعليمي عن الدليل التجريبي؟ ☐

☐ Did you separate the educational example from experimental evidence?

32.4 LLM وRAG في مشروع علمي

32.4 LLMs and RAG in a Scientific Project

في التوليد والاقتراح العلمي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «LLM وRAG في مشروع علمي»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific generation and proposal workflows, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “LLMs and RAG in a Scientific Project”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «LLM وRAG في مشروع علمي» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “LLMs and RAG in a Scientific Project” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```python
def approve_candidate(candidate, required_fields):
    missing = [k for k in required_fields if k not in candidate]
    if missing:
        raise ValueError(f"missing fields: {missing}")
    if candidate.get("evidence_status") != "reviewed":
        return False
    return True
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

32.5 التحقق من المرشحين المولّدين

32.5 Validating Generated Candidates

في التوليد والاقتراح العلمي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «التحقق من المرشحين المولّدين»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific generation and proposal workflows, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Validating Generated Candidates”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column

order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «التحقق من المرشحين المولّدين» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Validating Generated Candidates” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

32.6 نظام توصية قابل للتفسير

32.6 An Explainable Recommendation System

في التوليد والاقتراح العلمي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «نظام توصية قابل للتفسير»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific generation and proposal workflows, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or

reviewer to inspect it. Applied to “An Explainable Recommendation System”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «نظام توصية قابل للتفسير» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “An Explainable Recommendation System” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
from sklearn.neighbors import NearestNeighbors
from sklearn.preprocessing import StandardScaler

Xz = StandardScaler().fit_transform(X)
model = NearestNeighbors(n_neighbors=5).fit(Xz)
distance, index = model.kneighbors(target.reshape(1, -1))
print(index[0], distance[0])
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

☐ هل حددت المدخلات والوحدات والإصدار قبل الحساب؟

☐ Were the inputs, units, and version defined before computation?

☐ هل يمكن إعادة بناء النتيجة من السجل نفسه؟

☐ Can the result be reconstructed from the same record?

☐ هل فصلت المثال التعليمي عن الدليل التجريبي؟

☐ Did you separate the educational example from experimental evidence?

32.7 سجل الاقتراحات وسبب القرار

32.7 Proposal Logs and Decision Rationale

في التوليد والاقتراح العلمي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «سجل الاقتراحات وسبب القرار»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific generation and proposal workflows, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires,

what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Proposal Logs and Decision Rationale”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «سجل الاقتراحات وسبب القرار» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Proposal Logs and Decision Rationale” must be stated with its limits: what was actually computed, what is an

educational example, and what requires independent data or experiment before it can be treated as evidence.

32.8 منع الخلط بين التوليد والاكتشاف العلمي

32.8 Preventing the Confusion Between Generation and Scientific Discovery

في التوليد والاقتراح العلمي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «منع الخلط بين التوليد والاكتشاف العلمي»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific generation and proposal workflows, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Preventing the Confusion Between Generation and Scientific Discovery”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative

overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «منع الخلط بين التوليد والاكتشاف العلمي» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Preventing the Confusion Between Generation and Scientific Discovery” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
def approve_candidate(candidate, required_fields):
    missing = [k for k in required_fields if k not in candidate]
    if missing:
        raise ValueError(f"missing fields: {missing}")
    if candidate.get("evidence_status") != "reviewed":
        return False
    return True
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

PYTHON

```
from dataclasses import dataclass

@dataclass(frozen=True)
class Proposal:
    text: str
    source: str | None
    status: str = "pending"

p = Proposal("candidate generated by an LLM", None)
print(p.status)
```

يمكن استخدام نموذج لغوي كبير لتحويل سؤال الباحث إلى استعلام منظم أو تلخيص سجل تجربة، لكن المخرج لا يدخل قاعدة البيانات بوصفه حقيقة. يطلب النظام بنية محددة، يتحقق منها، ويربط كل قيمة بمصدر أو سجل تشغيل. وإذا اقترح النموذج عنصرًا أو مركبًا أو تفسيرًا، يبقى الاقتراح في حالة pending حتى يمر بقواعد التحقق والموافقة. هكذا يكون الذكاء الاصطناعي طبقة مساعدة في الواجهة لا سلطة مستقلة على البيانات.

A large language model can convert a researcher question into a structured query or summarize an experiment record, but its output must not enter the database as established fact. The system requests a defined structure, validates it, and links fields to a source or run record. If the model proposes an element, compound, or interpretation, the proposal remains pending until validation and approval. AI therefore assists the interface without becoming an independent authority over scientific data.

32.10 LLM مع مخرجات منظمة ومراجعة

32.10 LLMs with Structured Output and Review

PYTHON

```
def validate_candidate(x, bounds):
    return all(
        bounds[k][0] <= x[k] <= bounds[k][1]
        for k in bounds
    )

candidate = {"density": 5.2, "melting_point": 800.0}
bounds = {"density": (0.0, 25.0),
          "melting_point": (0.0, 4000.0)}
print(validate_candidate(candidate, bounds))
```

في VAE لا يكفي عرض عينات مولدة جميلة. يجب تعريف مجموعة البيانات، وطريقة التقسيم، وما الذي يتعلمه الفضاء الكامن، وما المقياس الذي يحدد جودة إعادة البناء والعينات. مع 118 سجلًا فقط تكون النماذج العميقة محدودة القيمة؛ لكن يمكن استخدام VAE لتعليم مفهوم التمثيل الكامن مع التحقق من أن المخرجات تقع ضمن نطاقات فيزيائية معقولة. العينة غير الصالحة تسجل كرفض ولا تتحول إلى «اكتشاف».

For a VAE, visually attractive samples are not enough. Define the dataset, split, latent representation, and metrics for reconstruction and generated outputs. With only 118 records, deep models have limited scientific value, but a VAE can still teach latent representations if outputs are checked against physically meaningful ranges. Invalid candidates should be recorded as rejected outputs rather than presented as discoveries.

32.9 VAE كمولد قابل للقياس

32.9 A VAE as a Measurable Generator

الفصل 33: المحاكاة الجزيئية والكمية — H_2 وH_2O وCO_2 و4–6 كيوبت

Chapter 33: Molecular and Quantum Simulation — H_2, H_2O, CO_2, and 4–6 Qubits

هذا الفصل يضيف طبقة عملية إلى المشروع الحالي. الفكرة ليست زيادة عدد الأدوات، بل جعل المسار من البيانات إلى النتيجة أكثر وضوحًا وقابلية للقياس والمراجعة. كل مثال هنا يستخدم بيانات المشروع أو بيانات تعليمية مصطنعة بوضوح، ولا يعرض نتيجة حسابية على أنها حقيقة تجريبية لم تُختبر.

This chapter adds an operational layer to the existing project. The goal is not to increase the number of tools, but to make the path from data to result clearer, measurable, and reviewable. Each example uses project data or explicitly synthetic educational data,

and no computational result is presented as experimental fact unless it has been independently tested.

33.1 من الهيدروجين إلى جزيئات أكبر

33.1 From Hydrogen to Larger Molecules

في الكيمياء الحاسوبية والحوسبة الكمومية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «من الهيدروجين إلى جزيئات أكبر»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In computational chemistry and quantum computing, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “From Hydrogen to Larger Molecules”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «من الهيدروجين إلى جزيئات أكبر» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “From Hydrogen to Larger Molecules” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

33.2 الهندسة الجزيئية وقاعدة الأساس

33.2 Molecular Geometry and Basis Sets

في الكيمياء الحاسوبية والحوسبة الكمومية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «الهندسة الجزيئية وقاعدة الأساس»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In computational chemistry and quantum computing, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Molecular Geometry and Basis Sets”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو

دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «الهندسة الجزيئية وقاعدة الأساس» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Molecular Geometry and Basis Sets” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
from pyscf import gto, scf

mol = gto.M(
    atom="O 0 0 0; H 0 0.757; H 0 -0.757",
    basis="sto-3g",
    unit="Angstrom",
)
mf = scf.RHF(mol).run()
print("H2O RHF energy:", mf.e_tot)
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

33.3 عدد الإلكترونات ليس عدد الكيوبتات

33.3 Electron Count Is Not the Same as Qubit Count

في الكيمياء الحاسوبية والحوسبة الكمومية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «عدد الإلكترونات ليس عدد الكيوبتات»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In computational chemistry and quantum computing, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Electron Count Is Not the Same as Qubit Count”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column

order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «عدد الإلكترونات ليس عدد الكيوبتات» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Electron Count Is Not the Same as Qubit Count” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

هل حددت المدخلات والوحدات والإصدار قبل الحساب؟ ☐

☐ Were the inputs, units, and version defined before computation?

هل يمكن إعادة بناء النتيجة من السجل نفسه؟ ☐

☐ Can the result be reconstructed from the same record?

هل فصلت المثال التعليمي عن الدليل التجريبي؟ ☐

☐ Did you separate the educational example from experimental evidence?

33.4 H_2O كمثال تعليمي منضبط

33.4 H_2O as a Controlled Educational Example

في الكيمياء الحاسوبية والحوسبة الكمومية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «H_2O كمثال تعليمي منضبط»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In computational chemistry and quantum computing, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “H_2O as a Controlled Educational Example”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة

كمثال تعليمي منضبط» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو H_2O«
تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of "H_2O as a Controlled Educational Example" must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
from pyscf import gto, scf

mol = gto.M(
    atom="O 0 0 0; C 0 0 1.16; O 0 0 2.32",
    basis="sto-3g",
    unit="Angstrom",
)
energy = scf.RHF(mol).run().e_tot
print("CO2 RHF reference energy:", energy)
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

33.5 CO_2 ولماذا تزداد الكلفة الحسابية

33.5 CO_2 and Why Computational Cost Increases

في الكيمياء الحاسوبية والحوسبة الكمومية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع

ولماذا تزداد الكلفة الحسابية»، تصبح العملية سلسلة واضحة من افتراضات وبيانات CO_2» بفحصه. عند تطبيق ذلك على وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In computational chemistry and quantum computing, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “CO_2 and Why Computational Cost Increases”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة ولماذا تزداد الكلفة الحسابية» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات CO_2» أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal

may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “CO_2 and Why Computational Cost Increases” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

33.6 VQE وحلقة التحسين الهجينة

33.6 VQE and the Hybrid Optimization Loop

في الكيمياء الحاسوبية والحوسبة الكمومية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «VQE وحلقة التحسين الهجينة»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In computational chemistry and quantum computing, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “VQE and the Hybrid Optimization Loop”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a

quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «VQE وحلقة التحسين الهجينة» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “VQE and the Hybrid Optimization Loop” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
from pyscf import gto, scf

mol = gto.M(
    atom="O 0 0 0; H 0 0.757; H 0 -0.757",
    basis="sto-3g",
    unit="Angstrom",
)
mf = scf.RHF(mol).run()
print("H2O RHF energy:", mf.e_tot)
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded.

The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

هل حددت المدخلات والوحدات والإصدار قبل الحساب؟ ☐

☐ Were the inputs, units, and version defined before computation?

هل يمكن إعادة بناء النتيجة من السجل نفسه؟ ☐

☐ Can the result be reconstructed from the same record?

هل فصلت المثال التعليمي عن الدليل التجريبي؟ ☐

☐ Did you separate the educational example from experimental evidence?

33.7 HHL وحدوده العلمية

33.7 HHL and Its Scientific Limits

في الكيمياء الحاسوبية والحوسبة الكمومية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «HHL وحدوده العلمية»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In computational chemistry and quantum computing, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “HHL and Its Scientific Limits”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «HHL وحدوده العلمية» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “HHL and Its Scientific Limits” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

33.8 ميزانية الموارد قبل تشغيل QPU

33.8 Resource Budgeting Before Running a QPU

في الكيمياء الحاسوبية والحوسبة الكمومية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «ميزانية الموارد قبل تشغيل QPU»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In computational chemistry and quantum computing, the goal should not be to add a technical layer merely because it exists. The

first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Resource Budgeting Before Running a QPU”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات «QPU «ميزانية الموارد قبل تشغيل أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Resource Budgeting Before Running a QPU” must be stated with its limits: what was actually computed, what is

an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
from pyscf import gto, scf

mol = gto.M(
    atom="O 0 0 0; C 0 0 1.16; O 0 0 2.32",
    basis="sto-3g",
    unit="Angstrom",
)
energy = scf.RHF(mol).run().e_tot
print("CO2 RHF reference energy:", energy)
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

PYTHON / classical_baseline_h2.py

```
from __future__ import annotations
from pyscf import gto, scf, fci

mol = gto.M(atom="H 0 0 0; H 0 0 0.74", basis="sto-3g", unit="Angstrom", charge=0, spin=0)
mf = scf.RHF(mol)
energy_hf = mf.kernel()
fci_solver = fci.FCI(mol, mf.mo_coeff)
energy_fci, _ = fci_solver.kernel()
print(f"Hartree-Fock energy: {energy_hf:.10f} Ha")
print(f"FCI energy:          {energy_fci:.10f} Ha")
print(f"FCI - HF:            {energy_fci - energy_hf:.10f} Ha")
```

الطاقة المحسوبة ليست مستقلة عن هندسة الجزيء. إذا كانت المسافة بين الذرات غير مناسبة، فقد تكون المقارنة بين VQE ومرجع كلاسيكي مضللة. في المثال التعليمي يمكن تثبيت هندسة H_2، ثم تجربة مسافات قليلة ومراقبة منحنى الطاقة. الهدف هو تعليم القارئ أن نتيجة الخوارزمية مرتبطة أيضًا بالمدخل الفيزيائي، وليس بالكود وحده.

Computed energy depends on molecular geometry. If the interatomic distance is inappropriate, a comparison between VQE

and a classical reference can be misleading. In the educational example, the H_2 geometry can be fixed and several bond lengths sampled to observe the energy curve. This teaches that an algorithmic result depends on the physical input, not only on the code.

33.11 اختبار التقارب الهندسي

33.11 Geometry-Convergence Testing

عند الانتقال من H_2 إلى جزيء أكبر، يصبح اختيار قاعدة الأساس والفضاء النشط جزءًا من تعريف المسألة نفسها. قاعدة أساس صغيرة تقلل الكلفة لكنها قد تفقد بعض الدقة، بينما زيادة عدد المدارات ترفع عدد المتغيرات والكيوبتات. لذلك يجب أن يسجل المشروع اسم قاعدة الأساس، وعدد المدارات، وعدد الإلكترونات، وأي تقليل طُبق قبل بناء الدائرة.

When moving from H_2 to a larger molecule, the basis set and active-space choice become part of the problem definition. A small basis reduces cost but can lose accuracy, while adding orbitals increases the number of variables and qubits. The project should therefore record the basis set, number of orbitals, number of electrons, and any reduction applied before constructing the circuit.

33.10 اختيار قاعدة الأساس والفضاء النشط

33.10 Basis-Set Choice and Active-Space Selection

PYTHON

```python
import numpy as np

# Classical reference for Ax=b.
A = np.array([[1.0, 0.0], [0.0, 3.0]])
b = np.array([1.0, 1.0])
x = np.linalg.solve(A, b)

print("classical reference:", x)
print("normalized:", x / np.linalg.norm(x))

# The HHL circuit is a separate artifact whose
# phase precision, controlled rotation, and
# post-selection assumptions must be recorded.
```

لإدخال HHL في التنفيذ دون ادعاء ما لا يفعله، نستخدم نظامًا خطيًا صغيرًا محددًا ونقارن الحل الكمومي التعليمي بالحل الكلاسيكي. نختار مصفوفة Hermitian بسيطة ذات قيم ذاتية معروفة، ثم تُبنى مراحل تحضير الحالة وتقدير الطور والدوران المشروط والقياس. الهدف ليس إثبات تفوق كمومي؛ المثال يثبت أن HHL يمكن تمثيله كخوارزمية لأنظمة خطية عندما تكون الفرضيات واضحة. تحفظ المصفوفة والحالة والمعلمات والمرجع الكلاسيكي مع سجل التجربة.

To include HHL operationally without overstating its scope, use a small defined linear system and compare the educational quantum construction with a classical solution. Choose a simple Hermitian matrix with known eigenvalues, then construct state preparation, phase estimation, controlled rotation, and measurement stages. The goal is not to claim quantum advantage; it is to demonstrate HHL as a linear-system algorithm when its assumptions are explicit. Store the matrix, state, parameters, and classical reference with the experiment record.

33.10 HHL كتطبيق فعلي لنظام خطي صغير

33.10 HHL as a Practical Small Linear-System Example

PYTHON

```python
from qiskit_nature.units import DistanceUnit
from qiskit_nature.second_q.drivers import PySCFDriver
from qiskit_nature.second_q.transformers import ActiveSpaceTransformer

molecules = {
    "H2O": "O 0 0 0; H 0.758 0 0.504; H -0.758 0 0.504",
    "CO2": "O 0 0 0; C 0 0 1.16; O 0 0 2.32",
}

for name, atom in molecules.items():
    full = PySCFDriver(
        atom=atom, basis="sto3g",
        unit=DistanceUnit.ANGSTROM
    ).run()
    problem = ActiveSpaceTransformer(2, 2).transform(full)
    print(name, problem.num_particles,
          problem.num_spatial_orbitals)
```

يمكن تنفيذ H_2O وCO_2 بدل تركهما كعناوين نظرية بشرط أن نكون صريحين بشأن الاختزال. يبني PySCF المسألة الإلكترونية، ثم يختار ActiveSpaceTransformer فضاءً صغيرًا من إلكترونين ومدارين مكانيين حتى تصبح التجربة قابلة للتشغيل على محاكي. هذا ليس تمثيلًا كاملًا لكل

إلكترونات الجزيء؛ إنه نموذج محدد يمكن مقارنته بمرجع كلاسيكي. قيمة المثال أن القارئ يرى كيف تتغير المشكلة عندما تتغير الهندسة وعدد العناصر، مع بقاء سجل التجربة واضحًا.

H_2O and CO_2 can be executed rather than left as theoretical headings if the reduction is explicit. PySCF builds the electronic-structure problem, and ActiveSpaceTransformer selects a small space of two electrons and two spatial orbitals so the experiment can run on a simulator. This is not a complete representation of every electron; it is a defined model that can be compared with a classical reference. The reader can therefore see how geometry and composition change the problem while the experiment record remains explicit.

33.9 تنفيذ H_2O وCO_2 في فضاء نشط محدد

33.9 Executing H_2O and CO_2 in a Defined Active Space

الفصل 34: هندسة العتاد الكمومي — الضوضاء والتحويل والتنفيذ السحابي

Chapter 34: Quantum Hardware Engineering — Noise, Transpilation, and Cloud Execution

هذا الفصل يضيف طبقة عملية إلى المشروع الحالي. الفكرة ليست زيادة عدد الأدوات، بل جعل المسار من البيانات إلى النتيجة أكثر وضوحًا وقابلية للقياس والمراجعة. كل مثال هنا يستخدم بيانات المشروع أو بيانات تعليمية مصطنعة بوضوح، ولا يعرض نتيجة حسابية على أنها حقيقة تجريبية لم تُختبر.

This chapter adds an operational layer to the existing project. The goal is not to increase the number of tools, but to make the path from data to result clearer, measurable, and reviewable. Each example uses project data or explicitly synthetic educational data, and no computational result is presented as experimental fact unless it has been independently tested.

المحاكاة المثالية ليست جهازًا حقيقيًا 34.1

34.1 An Ideal Simulation Is Not Real Hardware

في العتاد الكمومي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «المحاكاة المثالية ليست جهازًا حقيقيًا»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In quantum-hardware workflows, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “An Ideal Simulation Is Not Real Hardware”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة

«المحاكاة المثالية ليست جهازًا حقيقيًا» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “An Ideal Simulation Is Not Real Hardware” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

34.2 القياس وعدد اللقطات

34.2 Measurement and Shot Count

في العتاد الكمومي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «القياس وعدد اللقطات»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In quantum-hardware workflows, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Measurement and Shot Count”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «القياس وعدد اللقطات» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Measurement and Shot Count” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```python
from qiskit import QuantumCircuit

qc = QuantumCircuit(2)
qc.h(0)
qc.cx(0, 1)
qc.measure_all()
print(qc)
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

34.3 التحويل إلى ISA circuits

34.3 Transpiling to ISA Circuits

في العتاد الكمومي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «التحويل إلى ISA circuits»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In quantum-hardware workflows, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Transpiling to ISA Circuits”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software

versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «التحويل إلى ISA circuits» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Transpiling to ISA Circuits” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

☐ هل حددت المدخلات والوحدات والإصدار قبل الحساب؟

☐ Were the inputs, units, and version defined before computation?

☐ هل يمكن إعادة بناء النتيجة من السجل نفسه؟

☐ Can the result be reconstructed from the same record?

☐ هل فصلت المثال التعليمي عن الدليل التجريبي؟

☐ Did you separate the educational example from experimental evidence?

34.4 EstimatorV2 وSamplerV2

34.4 EstimatorV2 and SamplerV2

في العتاد الكمومي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق

تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، «EstimatorV2 وSamplerV2» ذلك على بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In quantum-hardware workflows, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “EstimatorV2 and SamplerV2”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج «EstimatorV2 وSamplerV2» بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an

ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “EstimatorV2 and SamplerV2” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```python
from qiskit_ibm_runtime import QiskitRuntimeService, SamplerV2

service = QiskitRuntimeService()
backend = service.least_busy(operational=True, simulator=False)
sampler = SamplerV2(mode=backend)
# job = sampler.run([(qc,)], shots=1024)
# result = job.result()
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

34.5 الجلسات والتنفيذ التكراري

34.5 Sessions and Iterative Execution

في العتاد الكمومي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «الجلسات والتنفيذ التكراري»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In quantum-hardware workflows, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Sessions and Iterative Execution”, this turns the

workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «الجلسات والتنفيذ التكراري» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Sessions and Iterative Execution” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

34.6 Amazon Braket Hybrid Jobs

34.6 Amazon Braket Hybrid Jobs

في العتاد الكمومي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «Amazon Braket Hybrid Jobs»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In quantum-hardware workflows, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Amazon Braket Hybrid Jobs”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة

مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج «Amazon Braket Hybrid Jobs» بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Amazon Braket Hybrid Jobs” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
from qiskit import QuantumCircuit

qc = QuantumCircuit(2)
qc.h(0)
qc.cx(0, 1)
qc.measure_all()
print(qc)
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

☐ هل حددت المدخلات والوحدات والإصدار قبل الحساب؟

☐ Were the inputs, units, and version defined before computation?

☐ هل يمكن إعادة بناء النتيجة من السجل نفسه؟

☐ Can the result be reconstructed from the same record?

☐ هل فصلت المثال التعليمي عن الدليل التجريبي؟

☐ Did you separate the educational example from experimental evidence?

تتبع الجهاز والنسخة والضوضاء 34.7

34.7 Tracking Device, Version, and Noise

في العتاد الكمومي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «تتبع الجهاز والنسخة والضوضاء»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In quantum-hardware workflows, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Tracking Device, Version, and Noise”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «تتبع الجهاز والنسخة والضوضاء» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Tracking Device, Version, and Noise” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

34.8 سياسة الاختبار قبل العتاد الحقيقي

34.8 A Test Policy Before Real Hardware

في العتاد الكمومي لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «سياسة الاختبار قبل العتاد الحقيقي»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In quantum-hardware workflows, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “A Test Policy Before Real Hardware”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو

دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «سياسة الاختبار قبل العتاد الحقيقي» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “A Test Policy Before Real Hardware” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```python
from qiskit_ibm_runtime import QiskitRuntimeService, SamplerV2

service = QiskitRuntimeService()
backend = service.least_busy(operational=True, simulator=False)
sampler = SamplerV2(mode=backend)
# job = sampler.run([(qc,)], shots=1024)
# result = job.result()
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

واجهات Runtime الحديثة تتطلب أن تكون الدائرة والمرصودات متوافقة مع تعريف الهدف قبل التنفيذ. لذلك يجب أن تكون عملية التحويل transpilation جزءًا مسجلًا من التجربة، لا خطوة خفية. يمكن حفظ عمق الدائرة وعدد البوابات وعدد الكيوبتات بعد التحويل لمقارنة نسختين من نفس الخوارزمية.

Modern Runtime interfaces require circuits and observables to conform to the target definition before execution. Transpilation should therefore be a recorded part of the experiment rather than a hidden step. Circuit depth, gate counts, and qubit counts after transpilation can be stored to compare two versions of the same algorithm.

34.10 ISA والتحويل قبل التنفيذ

34.10 ISA Compliance and Transpilation Before Execution

قبل إرسال دائرة إلى خدمة كمومية، يمكن اختبار البرنامج محليًا باستخدام محاكٍ أو backend وهمي متوافق مع واجهة Runtime. هذا الاختبار لا يثبت أن النتيجة على الجهاز الحقيقي ستكون نفسها، لكنه يكشف أخطاء في شكل الدائرة، والواجهة، وتسجيل النتائج قبل استهلاك موارد سحابية.

Before sending a circuit to a quantum service, the program can be tested locally with a simulator or a fake backend compatible with the Runtime interface. This does not prove that the hardware result will

match the local result, but it catches interface, circuit-shape, and result-logging errors before cloud resources are consumed.

34.9 الاختبار المحلي قبل الاتصال بالخدمة

34.9 Local Testing Before Cloud Execution

PYTHON

```
def summarize_runs(ideal, noisy, hardware):
    return {
        "ideal": float(ideal),
        "noisy": float(noisy),
        "hardware": float(hardware),
        "hardware_minus_ideal": float(hardware - ideal),
    }

print(summarize_runs(-1.13, -1.08, -1.04))
```

النتيجة من المحاكي المثالي ليست كافية للحكم على جهاز حقيقي. ينبغي أن يحتوي التقرير على خط أساس مثالي، ثم محاكاة ضوضائية إن أمكن، ثم نتيجة الجهاز. المقارنة تكشف أين تدخل الفروق: في التحويل، أو البوابات، أو القياس، أو تخفيف الخطأ. ويجب إعلان عدد shots لأن عدم اليقين الإحصائي يتغير معه. هكذا يصبح الادعاء عن تحسن الأداء نتيجة مقارنة محددة.

An ideal simulator result is not sufficient to judge real hardware. The report should contain an ideal baseline, a noisy simulation when available, and the hardware result. Comparing these layers helps identify whether discrepancies arise from transpilation, gates, measurement, or mitigation. The shot count should be declared because sampling uncertainty changes with it. A performance claim then becomes the result of a defined comparison.

34.10 الضوضاء والمرجع المثالي

34.10 Noise and the Ideal Reference

PYTHON

```
from qiskit_ibm_runtime import QiskitRuntimeService, EstimatorV2

service = QiskitRuntimeService()
backend = service.least_busy(
    operational=True, simulator=False
)
estimator = EstimatorV2(mode=backend)
```

```
record = {
    "backend": backend.name,
    "primitive": "EstimatorV2",
    "status": "created",
}
print(record)
```

عند الانتقال إلى خدمة كمومية يجب ألا يكون السجل مجرد رقم job. نحتاج إلى الخلفية، وإصدار البرنامج، وبنية الدائرة بعد التحويل، والمرصودات، وعدد اللقطات أو الدقة، وحالة المهمة، والنتيجة، وأي إعدادات لتخفيف الخطأ. واجهات EstimatorV2 وSamplerV2 الحالية تستخدم PUBs وتعيد RuntimeJobV2، لذلك يجب حفظ تعريف الإدخال والنتيجة معًا. إذا فشل الاتصال أو تغيرت الخلفية، يسجل الفشل بدل اختراع قيمة بديلة.

When moving to a quantum service, the record should be more than a job number. Store the backend, software version, transpiled circuit, observables, shots or requested precision, job status, result, and error-mitigation settings. Current EstimatorV2 and SamplerV2 interfaces use PUBs and return RuntimeJobV2 objects, so input definitions and results should be stored together. If the service call fails or the backend changes, record the failure rather than inventing a replacement value.

34.9 Recording a Qiskit Runtime V2 Execution

34.9 سجل تشغيل Qiskit Runtime V2

الفصل 35: الجسر الجينومي — العناصر والبروتينات والإنزيمات ومتتبع المقاومة

Chapter 35: The Genomic Bridge — Elements, Proteins, Enzymes, and Resistance Tracking

هذا الفصل يضيف طبقة عملية إلى المشروع الحالي. الفكرة ليست زيادة عدد الأدوات، بل جعل المسار من البيانات إلى النتيجة أكثر وضوحًا وقابلية للقياس والمراجعة. كل مثال هنا يستخدم بيانات المشروع أو بيانات تعليمية مصطنعة بوضوح، ولا يعرض نتيجة حسابية على أنها حقيقة تجريبية لم تُختبر.

This chapter adds an operational layer to the existing project. The goal is not to increase the number of tools, but to make the path from data to result clearer, measurable, and reviewable. Each example uses project data or explicitly synthetic educational data, and no computational result is presented as experimental fact unless it has been independently tested.

35.1 ما الذي يمكن ربطه فعلاً؟

35.1 What Can Actually Be Linked?

في البيانات الجينومية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «ما الذي يمكن ربطه فعلاً؟»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In genomic data systems, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to "What Can Actually Be Linked?", this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a

quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «ما الذي يمكن ربطه فعلاً؟» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of "What Can Actually Be Linked?" must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

35.2 مخطط البيانات الجينومي الإضافي

35.2 The Additional Genomic Data Schema

في البيانات الجينومية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «مخطط البيانات الجينومي الإضافي»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In genomic data systems, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to "The Additional Genomic Data Schema", this turns the workflow into an explicit chain of assumptions, data, computations, and results

rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «مخطط البيانات الجينومي الإضافي» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “The Additional Genomic Data Schema” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

SQL

```
CREATE TABLE genomic_links (
    link_id INTEGER PRIMARY KEY,
    atomic_number INTEGER NOT NULL,
```

```
    protein_id TEXT,
    enzyme_id TEXT,
    evidence_id TEXT NOT NULL,
    data_version TEXT NOT NULL,
    FOREIGN KEY (atomic_number) REFERENCES elements(atomic_number)
);
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

35.3 الأدلة ليست حقولًا تجميلية

35.3 Evidence Is Not Decorative Metadata

في البيانات الجينومية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «الأدلة ليست حقولًا تجميلية»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In genomic data systems, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Evidence Is Not Decorative Metadata”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «الأدلة ليست حقولًا تجميلية» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Evidence Is Not Decorative Metadata” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

هل حددت المدخلات والوحدات والإصدار قبل الحساب؟ ☐

☐ Were the inputs, units, and version defined before computation?

هل يمكن إعادة بناء النتيجة من السجل نفسه؟ ☐

☐ Can the result be reconstructed from the same record?

هل فصلت المثال التعليمي عن الدليل التجريبي؟ ☐

☐ Did you separate the educational example from experimental evidence?

35.4 Gene Resistance Tracker كسجل أدلة

35.4 Gene Resistance Tracker as an Evidence Record

في البيانات الجينومية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «Gene Resistance Tracker كسجل أدلة»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In genomic data systems, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Gene Resistance Tracker as an Evidence Record”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة

كسجل أدلة» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي «Gene Resistance Tracker يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Gene Resistance Tracker as an Evidence Record” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
from dataclasses import dataclass

@dataclass(frozen=True)
class EvidenceLink:
    atomic_number: int
    protein_id: str
    evidence_id: str
    data_version: str
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

35.5 فصل الهوية عن البيانات الحساسة

35.5 Separating Identity From Sensitive Data

في البيانات الجينومية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «فصل الهوية عن البيانات الحساسة»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In genomic data systems, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Separating Identity From Sensitive Data”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «فصل الهوية عن البيانات الحساسة» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware.

Therefore, the result of “Separating Identity From Sensitive Data” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

35.6 ربط النسخة بمصدرها

35.6 Linking a Version to Its Source

في البيانات الجينومية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «ربط النسخة بمصدرها»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In genomic data systems, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Linking a Version to Its Source”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative

overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «ربط النسخة بمصدرها» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Linking a Version to Its Source” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

SQL

```sql
CREATE TABLE genomic_links (
    link_id INTEGER PRIMARY KEY,
    atomic_number INTEGER NOT NULL,
    protein_id TEXT,
    enzyme_id TEXT,
    evidence_id TEXT NOT NULL,
    data_version TEXT NOT NULL,
    FOREIGN KEY (atomic_number) REFERENCES elements(atomic_number)
);
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

هل حددت المدخلات والوحدات والإصدار قبل الحساب؟ ☐

☐ Were the inputs, units, and version defined before computation?

هل يمكن إعادة بناء النتيجة من السجل نفسه؟ ☐

☐ Can the result be reconstructed from the same record?

هل فصلت المثال التعليمي عن الدليل التجريبي؟ ☐

☐ Did you separate the educational example from experimental evidence?

35.7 الاستعلام العلمي دون ادعاء السببية

35.7 Scientific Querying Without Claiming Causality

في البيانات الجينومية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «الاستعلام العلمي دون ادعاء السببية»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In genomic data systems, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Scientific Querying Without Claiming Causality”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be

defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «الاستعلام العلمي دون ادعاء السببية» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Scientific Querying Without Claiming Causality” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

حدود الجسر بين الكيمياء والبيولوجيا 35.8

35.8 Limits of the Chemistry-Biology Bridge

في البيانات الجينومية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «حدود الجسر بين الكيمياء والبيولوجيا»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In genomic data systems, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to

“Limits of the Chemistry-Biology Bridge”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «حدود الجسر بين الكيمياء والبيولوجيا» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Limits of the Chemistry-Biology Bridge” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
from dataclasses import dataclass

@dataclass(frozen=True)
class EvidenceLink:
    atomic_number: int
    protein_id: str
    evidence_id: str
    data_version: str
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

إذا ظهرت علاقة بين وفرة عنصر ومؤشر مرضي، فهذا لا يكفي لإثبات السببية. يجب النظر في تصميم الدراسة، والمتغيرات المربكة، والتحيز، وتسلسل الزمن، ونوعية الدليل. لذلك يسجل Gene Resistance Tracker نوع الدليل ومصدره ووقت الملاحظة، ولا يحول مجرد وجود علاقة في SQL إلى حكم طبي.

An observed relationship between elemental abundance and a disease indicator is not enough to establish causality. Study design, confounding variables, bias, temporal order, and evidence quality must be considered. The Gene Resistance Tracker therefore records evidence type, source, and observation time rather than turning a database association into a medical claim.

35.9 الفرق بين الارتباط والسببية

35.9 Association Is Not Causation

عند بناء الجسر الجينومي لا يلزم إدخال بيانات صحية شخصية إلى المثال. يمكن البدء ببيانات عامة أو مجمعة أو اصطناعية، ثم إثبات أن المخطط يستطيع تمثيل gene وvariant وevidence

دون الاحتفاظ بهوية شخص. إذا استُخدمت بيانات بشرية حقيقية، يجب أن تكون شروط sampleو الوصول والخصوصية جزءًا من سجل البيانات نفسه.

A genomic bridge does not require personal health data in the educational example. Public, aggregated, or synthetic data can demonstrate that the schema represents gene, variant, evidence, and sample without storing a person’s identity. If real human data are used, access and privacy conditions should be part of the dataset record itself.

35.8 البيانات العامة والبيانات الصحية المجمعة

35.8 Public and Aggregated Health Data

SQL

```
CREATE TABLE genomic_observation (
    observation_id INTEGER PRIMARY KEY,
    sample_id TEXT NOT NULL,
    reference_version TEXT NOT NULL,
    variant_id TEXT NOT NULL,
    evidence_source TEXT NOT NULL,
    evidence_level TEXT NOT NULL,
    pipeline_version TEXT NOT NULL
);
```

في نظام جينومي حقيقي لا يكفي تخزين sequence أو variant وحده. يجب أن يرتبط السجل بالعينة وإصدار المرجع ومصدر البيانات وطريقة المعالجة ووقت التشغيل. يوضح Gene Resistance Tracker كيف تصبح الهوية والمصدر والإصدار ومستوى الدليل حقولًا إلزامية. أما التفسير السريري أو السببي فيبقى خارج هذه الطبقة ما لم يوجد دليل مستقل مناسب. بذلك يبقى تمثيل البيانات منفصلًا عن الحكم الطبي.

In a real genomic system, storing a sequence or variant alone is insufficient. A record should link to the sample, reference version, data source, processing method, and run time. Gene Resistance Tracker demonstrates how identity, provenance, version, and evidence level can become required fields. Clinical or causal interpretation remains outside this layer unless appropriate

independent evidence exists. Data representation therefore remains separate from medical judgment.

35.10 سلامة الهوية والمصدر في البيانات الجينومية

35.10 Identity and Provenance in Genomic Data

PYTHON

```python
import numpy as np
import pandas as pd

n = 2_000_000
rng = np.random.default_rng(42)
synthetic = pd.DataFrame({
    "sample_id": rng.integers(1, 100_000, n),
    "position": rng.integers(1, 250_000_000, n),
    "variant": rng.choice(
        ["A>G", "C>T", "G>C", "T>A"], n
    ),
})
print(synthetic.memory_usage(deep=True).sum() / 1024**2,
      "MiB")
```

الانتقال من 118 عنصرًا إلى حجم جينومي لا يعني إدخال جينوم شخص حقيقي في الكتاب. يمكن إنشاء بيانات اصطناعية تشبه شكل المسألة: ملايين السجلات، معرفات عينات، مواقع ومتغيرات، مع منعها من تمثيل فرد حقيقي. الهدف هو اختبار SQL وDask وSpark والتخزين على حجم أكبر بكثير من الجدول الدوري. يمكن توليد السجلات تدريجيًا وقياس الذاكرة وزمن القراءة والربط، فيرى القارئ لماذا تتغير هندسة النظام عند الانتقال إلى Big Data.

Moving from 118 elements to genome-scale data does not require placing a real human genome in the book. Synthetic data can reproduce the computational shape: millions of records, sample identifiers, positions, and variants, without representing a real person. The purpose is to benchmark SQL, Dask, Spark, and storage at a scale far beyond the periodic table. Generate records incrementally and measure memory, read time, and joins so the reader sees why architecture changes at Big Data scale.

35.9 نموذج جينوم بشري اصطناعي لاختبار Big Data

35.9 A Synthetic Human-Genome-Scale Model for Big-Data Testing

الفصل 36: الحوكمة قبل التوليد — الموافقة المسبقة وسجل التدقيق

Chapter 36: Governance Before Generation — Pre-Approval and Audit Trails

هذا الفصل يضيف طبقة عملية إلى المشروع الحالي. الفكرة ليست زيادة عدد الأدوات، بل جعل المسار من البيانات إلى النتيجة أكثر وضوحًا وقابلية للقياس والمراجعة. كل مثال هنا يستخدم بيانات المشروع أو بيانات تعليمية مصطنعة بوضوح، ولا يعرض نتيجة حسابية على أنها حقيقة تجريبية لم تُختبر.

This chapter adds an operational layer to the existing project. The goal is not to increase the number of tools, but to make the path from data to result clearer, measurable, and reviewable. Each example uses project data or explicitly synthetic educational data, and no computational result is presented as experimental fact unless it has been independently tested.

36.1 لماذا تكون الحوكمة جزءًا من النظام؟

36.1 Why Governance Belongs Inside the System

في الحوكمة العلمية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «لماذا تكون الحوكمة جزءًا من النظام؟»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific governance, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Why Governance Belongs Inside the System”, this turns the workflow into an explicit chain of assumptions, data, computations,

and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «لماذا تكون الحوكمة جزءًا من النظام؟» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Why Governance Belongs Inside the System” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

تصنيف مستوى المخاطر 36.2

36.2 Risk-Level Classification

في الحوكمة العلمية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «تصنيف مستوى المخاطر»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific governance, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Risk-Level Classification”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «تصنيف مستوى المخاطر» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Risk-Level Classification” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```python
def requires_preapproval(request):
    high_impact = request["impact_level"] >= 3
    uses_sensitive_data = request["sensitive_data"]
    return high_impact or uses_sensitive_data

if requires_preapproval(request):
    status = "PENDING_HUMAN_REVIEW"
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

36.3 بوابة الموافقة المسبقة

36.3 The Pre-Approval Gate

في الحوكمة العلمية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «بوابة الموافقة المسبقة»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific governance, the goal should not be to add a technical layer merely because it exists. The first questions are what problem

the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “The Pre-Approval Gate”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «بوابة الموافقة المسبقة» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “The Pre-Approval Gate” must be stated with its limits: what was actually computed, what is an educational

example, and what requires independent data or experiment before it can be treated as evidence.

هل حددت المدخلات والوحدات والإصدار قبل الحساب؟ ☐

☐ Were the inputs, units, and version defined before computation?

هل يمكن إعادة بناء النتيجة من السجل نفسه؟ ☐

☐ Can the result be reconstructed from the same record?

هل فصلت المثال التعليمي عن الدليل التجريبي؟ ☐

☐ Did you separate the educational example from experimental evidence?

36.4 سجل التدقيق غير القابل للالتباس

36.4 An Unambiguous Audit Log

في الحوكمة العلمية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «سجل التدقيق غير القابل للالتباس»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific governance, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “An Unambiguous Audit Log”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «سجل التدقيق غير القابل للالتباس» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “An Unambiguous Audit Log” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

SQL

```
CREATE TABLE audit_log (
    event_id INTEGER PRIMARY KEY,
    request_id TEXT NOT NULL,
    event_type TEXT NOT NULL,
    actor TEXT NOT NULL,
    event_time TEXT NOT NULL,
    payload_hash TEXT NOT NULL
);
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

36.5 المراجعة البشرية والاستثناءات

36.5 Human Review and Exceptions

في الحوكمة العلمية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «المراجعة البشرية والاستثناءات»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific governance, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Human Review and Exceptions”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software

versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «المراجعة البشرية والاستثناءات» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Human Review and Exceptions” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

فصل التنبؤ عن القرار 36.6

36.6 Separating Inference From Decision

في الحوكمة العلمية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «فصل التنبؤ عن القرار»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific governance, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Separating Inference From Decision”, this turns the workflow into an explicit chain of assumptions, data, computations, and results

rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «فصل التنبؤ عن القرار» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Separating Inference From Decision” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
def requires_preapproval(request):
    high_impact = request["impact_level"] >= 3
    uses_sensitive_data = request["sensitive_data"]
```

```
    return high_impact or uses_sensitive_data

if requires_preapproval(request):
    status = "PENDING_HUMAN_REVIEW"
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

☐ هل حددت المدخلات والوحدات والإصدار قبل الحساب؟

☐ Were the inputs, units, and version defined before computation?

☐ هل يمكن إعادة بناء النتيجة من السجل نفسه؟

☐ Can the result be reconstructed from the same record?

☐ هل فصلت المثال التعليمي عن الدليل التجريبي؟

☐ Did you separate the educational example from experimental evidence?

36.7 إيقاف التشغيل الآمن

36.7 Safe Shutdown and Refusal

في الحوكمة العلمية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «إيقاف التشغيل الآمن»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific governance, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to

“Safe Shutdown and Refusal”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «إيقاف التشغيل الآمن» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Safe Shutdown and Refusal” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

قياس الامتثال 36.8

36.8 Measuring Compliance

في الحوكمة العلمية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «قياس الامتثال»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific governance, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Measuring Compliance”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «قياس الامتثال» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Measuring Compliance” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

SQL

```sql
CREATE TABLE audit_log (
    event_id INTEGER PRIMARY KEY,
    request_id TEXT NOT NULL,
    event_type TEXT NOT NULL,
    actor TEXT NOT NULL,
    event_time TEXT NOT NULL,
    payload_hash TEXT NOT NULL
);
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

SQL / preapproval.sql

```sql
CREATE TABLE IF NOT EXISTS preapproval (
    approval_id TEXT PRIMARY KEY,
    purpose TEXT NOT NULL,
    data_scope TEXT NOT NULL,
    model_version TEXT NOT NULL,
    risk_class TEXT NOT NULL,
    reviewer TEXT NOT NULL,
    decision TEXT NOT NULL CHECK (decision IN ('approved', 'rejected', 'expired')),
    expires_at TEXT NOT NULL
);

-- A run must reference an approved record whose expiry has not passed.
```

في النظام القابل للمراجعة لا ينبغي أن يكون الشخص الذي يغيّر البيانات هو نفسه الشخص الوحيد الذي يعتمد النتيجة الحساسة. يمكن فصل أدوار data steward وanalyst وreviewer وrelease manager حتى في مشروع صغير، ولو عبر حقول واضحة في سجل التشغيل. هذا يقلل خطر الخطأ غير المكتشف ويجعل المسؤولية قابلة للتتبع.

In an auditable system, the person who changes data should not be the only person who approves a sensitive result. Roles such as data steward, analyst, reviewer, and release manager can be separated even in a small project through explicit fields in the run record. This reduces undetected error and makes responsibility traceable.

36.10 فصل المهام والمسؤوليات

36.10 Separation of Duties

يمكن تصنيف الطلبات إلى مستويات تشغيلية وفق نطاق البيانات ونوع المخرج وإمكانية التأثير. لا يعني التصنيف أن كل عملية عالية المستوى خطرة بالضرورة؛ بل يحدد مقدار المراجعة المطلوبة قبل التشغيل. مثال بسيط: تحليل وصفي على بيانات عامة قد يحتاج مراجعة آلية، بينما اقتراح توليدي متصل ببيانات حساسة يحتاج موافقة بشرية صريحة وسجلًا أقوى.

Requests can be assigned operational risk classes based on data scope, output type, and potential impact. A higher class does not mean that every run is inherently harmful; it defines the review level required before execution. A descriptive analysis on public data may use automated checks, while a generative proposal involving sensitive data requires explicit human approval and stronger logging.

36.9 تصنيف المخاطر قبل التشغيل

36.9 Risk Classification Before Execution

SQL

```
CREATE TABLE audit_event (
    event_id INTEGER PRIMARY KEY,
```

```
    request_id TEXT NOT NULL,
    event_type TEXT NOT NULL,
    actor TEXT NOT NULL,
    created_at TEXT NOT NULL,
    payload_json TEXT NOT NULL
);
```

عندما يُرفض اقتراح أو يُعدل قبل التنفيذ يجب ألا يختفي سبب القرار. سجل التدقيق يربط request_id بالمدخلات والنسخة البرمجية والمراجع وقرار الموافقة والنتيجة. يمكن تخزين الأحداث في جدول append-only بحيث لا يعيد التطبيق العادي كتابة التاريخ. عند التحقق في نتيجة يستطيع المراجع الانتقال من الرقم النهائي إلى العملية التي أنتجته ثم إلى القرار الذي سمح بالتشغيل. هذا هو المعنى العملي للمساءلة.

When a proposal is rejected or modified before execution, the reason should not disappear. An audit trail links a request identifier to inputs, software version, references, approval, and final result. Events can be stored in an append-only table so ordinary application code cannot silently rewrite history. During review, an auditor can move from the final number to the operation that produced it and then to the decision that authorized execution. That is the practical meaning of accountability.

36.10 سجل التدقيق وسلسلة القرار

36.10 Audit Trail and Decision Chain

PYTHON

```python
from dataclasses import dataclass

@dataclass(frozen=True)
class Approval:
    purpose: str
    risk: str
    approver: str | None

    def executable(self):
        return bool(
            self.purpose and self.approver
            and self.risk in {"low", "medium", "high"}
        )

print(Approval("benchmark", "low", "reviewer-1").executable())
```

الموافقة المسبقة لا ينبغي أن تبقى فقرة في وثيقة أخلاقية. يمكن تحويلها إلى بوابة تشغيلية تمنع انتقال الطلب إلى التنفيذ قبل اكتمال الحقول المطلوبة. يحدد السجل صاحب الطلب والغرض والبيانات ومستوى المخاطر والقيود والموافق. إذا كان الطلب توليديًا أو كموميًا، يضاف ما الذي سيعتبر نتيجة وما الذي سيعتبر اقتراحًا. البوابة لا تحكم على صحة العلم؛ وظيفتها منع التشغيل غير الموثق وإجبار النظام على توفير معلومات المراجعة.

Pre-approval should not remain a paragraph in an ethics document. It can become an operational gate that prevents execution until required fields are complete. The record identifies requester, purpose, data, risk level, constraints, and approver. For generative or quantum requests, it can also state what counts as a result and what counts only as a proposal. The gate does not decide scientific truth; it prevents undocumented execution and supplies reviewers with the needed information.

36.9 بوابة الموافقة المسبقة كمنطق قابل للاختبار

36.9 Pre-Approval as Testable Logic

الفصل 37: التشغيل والإنتاج — Docker وCI/CD والمراقبة وأمن السلسلة

Chapter 37: Production Engineering — Docker, CI/CD, Observability, and Supply-Chain Security

هذا الفصل يضيف طبقة عملية إلى المشروع الحالي. الفكرة ليست زيادة عدد الأدوات، بل جعل المسار من البيانات إلى النتيجة أكثر وضوحًا وقابلية للقياس والمراجعة. كل مثال هنا يستخدم بيانات المشروع أو بيانات تعليمية مصطنعة بوضوح، ولا يعرض نتيجة حسابية على أنها حقيقة تجريبية لم تُختبر.

This chapter adds an operational layer to the existing project. The goal is not to increase the number of tools, but to make the path from data to result clearer, measurable, and reviewable. Each example uses project data or explicitly synthetic educational data,

and no computational result is presented as experimental fact unless it has been independently tested.

37.1 البيئة ليست تفصيلاً ثانويًا

37.1 The Environment Is Not a Secondary Detail

في تشغيل البرمجيات العلمية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «البيئة ليست تفصيلاً ثانويًا»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific software operations, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “The Environment Is Not a Secondary Detail”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «البيئة ليست تفصيلاً ثانويًا» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “The Environment Is Not a Secondary Detail” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

37.2 Docker لبناء بيئة متطابقة

37.2 Docker for a Consistent Environment

في تشغيل البرمجيات العلمية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «Docker لبناء بيئة متطابقة»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific software operations, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Docker for a Consistent Environment”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو

دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «Docker لبناء بيئة متطابقة» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Docker for a Consistent Environment” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

DOCKERFILE

```
FROM python:3.12-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY . .
CMD ["python", "run_pipeline.py", "--stage", "all"]
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

37.3 الاختبار المستمر قبل الدمج

37.3 Continuous Testing Before Integration

في تشغيل البرمجيات العلمية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «الاختبار المستمر قبل الدمج»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific software operations, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Continuous Testing Before Integration”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software

versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «الاختبار المستمر قبل الدمج» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Continuous Testing Before Integration” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

هل حددت المدخلات والوحدات والإصدار قبل الحساب؟ ☐

☐ Were the inputs, units, and version defined before computation?

هل يمكن إعادة بناء النتيجة من السجل نفسه؟ ☐

☐ Can the result be reconstructed from the same record?

هل فصلت المثال التعليمي عن الدليل التجريبي؟ ☐

☐ Did you separate the educational example from experimental evidence?

37.4 فحص التبعيات والملفات المثبتة

37.4 Dependency and Lockfile Inspection

في تشغيل البرمجيات العلمية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند

تطبيق ذلك على «فحص التبعيات والملفات المثبتة»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific software operations, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Dependency and Lockfile Inspection”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «فحص التبعيات والملفات المثبتة» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an

ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Dependency and Lockfile Inspection” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

YAML

```yaml
name: ci
on: [push, pull_request]
jobs:
  test:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-python@v5
        with: {python-version: '3.12'}
      - run: pip install -r requirements.txt
      - run: python -m pytest -q
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

37.5 المراقبة والسجلات والمقاييس

37.5 Logs, Metrics, and Observability

في تشغيل البرمجيات العلمية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «المراقبة والسجلات والمقاييس»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific software operations, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it

produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Logs, Metrics, and Observability”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «المراقبة والسجلات والمقاييس» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Logs, Metrics, and Observability” must be stated with its limits: what was actually computed, what is an

educational example, and what requires independent data or experiment before it can be treated as evidence.

إدارة الأسرار دون وضعها في الكود 37.6

37.6 Secret Management Without Putting Secrets in Code

في تشغيل البرمجيات العلمية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «إدارة الأسرار دون وضعها في الكود»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific software operations, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Secret Management Without Putting Secrets in Code”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «إدارة الأسرار دون وضعها في الكود» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of "Secret Management Without Putting Secrets in Code" must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

DOCKERFILE

```
FROM python:3.12-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY . .
CMD ["python", "run_pipeline.py", "--stage", "all"]
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

☐ هل حددت المدخلات والوحدات والإصدار قبل الحساب؟

☐ Were the inputs, units, and version defined before computation?

☐ هل يمكن إعادة بناء النتيجة من السجل نفسه؟

☐ Can the result be reconstructed from the same record?

هل فصلت المثال التعليمي عن الدليل التجريبي؟ ☐

☐ Did you separate the educational example from experimental evidence?

37.7 الاستعادة من الفشل

37.7 Recovery From Failure

في تشغيل البرمجيات العلمية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «الاستعادة من الفشل»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific software operations, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Recovery From Failure”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative

overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «الاستعادة من الفشل» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Recovery From Failure” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

37.8 سجل الإصدارات كجزء من النتيجة العلمية

37.8 Version History as Part of the Scientific Result

في تشغيل البرمجيات العلمية لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «سجل الإصدارات كجزء من النتيجة العلمية»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In scientific software operations, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Version History as Part of the Scientific Result”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «سجل الإصدارات كجزء من النتيجة العلمية» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Version History as Part of the Scientific Result” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

YAML

```yaml
name: ci
on: [push, pull_request]
jobs:
  test:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-python@v5
        with: {python-version: '3.12'}
```

```
      - run: pip install -r requirements.txt
      - run: python -m pytest -q
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

YAML / .github/workflows/ci.yml

```yaml
name: scientific-ci

on:
  push:
  pull_request:

jobs:
  test:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-python@v5
        with:
          python-version: "3.12"
      - run: python -m pip install --upgrade pip
      - run: pip install -r requirements.txt
      - run: python run_pipeline.py --stage database
      - run: python -m pytest -q
```

مفاتيح API وكلمات المرور لا توضع داخل الكود أو الكتاب. يستخدم المشروع متغيرات بيئة أو مدير أسرار، بينما تبقى ملفات الإعداد الخالية من الأسرار قابلة للنشر. كما يجب أن يختبر CI عدم وجود أنماط تشبه المفاتيح داخل المستودع. هذه قاعدة تشغيلية بسيطة لكنها تمنع خطأ نشر يمكن أن يفسد المشروع كله.

API keys and passwords must not be embedded in source code or the book. The project should use environment variables or a secret manager, while sanitized configuration files remain publishable. CI should also scan for key-like patterns. This simple operational rule prevents a release error that could compromise the entire project.

37.10 الأسرار والإعدادات

37.10 Secrets and Configuration

عند نشر مشروع علمي، لا يكفي ذكر أسماء المكتبات. يجب أن يكون من الممكن معرفة الإصدارات التي دخلت في البناء. يمكن إنشاء Software Bill of Materials وتخزين hashes للملفات مع حفظ ملف requirements أو lockfile. إذا تغيرت مكتبة أساسية، يعاد تشغيل الاختبارات بدل افتراض أن النتيجة قابلة للمقارنة.

For a scientific release, naming libraries is not enough. The build should identify the exact versions involved. A Software Bill of Materials and file hashes can be stored alongside requirements or a lockfile. If a core dependency changes, the test suite should be rerun rather than assuming that results remain comparable.

37.9 البصمة البرمجية و SBOM

37.9 Software Fingerprints and SBOM

PYTHON

```
import streamlit as st

st.title("Part III — Scientific Dashboard")
st.metric("Elements", 118)
st.caption("Educational example; not a clinical judgment.")

if st.button("Show run contract"):
    st.json({
        "run_id": "demo-001",
        "data_version": "elements-v1",
        "status": "verified",
    })
```

يمكن إضافة واجهة Streamlit تعرض عدد العناصر وحالة آخر تشغيل وزمن الاستعلام ونتائج النموذج. لكن الواجهة لا تخفي المصدر ولا تعرض احتمالًا على أنه حقيقة. كل بطاقة نتيجة تشير إلى run_id ونسخة البيانات، ويمكنها عرض الاستعلام أو ملف النتائج. تصبح الواجهة طبقة عرض فوق النظام ويمكن استبدالها لاحقًا دون تغيير قاعدة البيانات أو الخوارزميات.

A small Streamlit interface can display element count, latest run status, query duration, and model outputs. The interface should not hide provenance or turn a probability into a fact. Each result card

references a run identifier and dataset version, and can expose the underlying query or result file. The interface remains a presentation layer and can be replaced without changing the database or algorithms.

37.10 واجهة Streamlit للعرض لا للحكم العلمي

37.10 A Streamlit Interface for Display, Not Scientific Judgment

DOCKERFILE

```
FROM python:3.12-slim
WORKDIR /app
COPY requirements.txt .
RUN python -m pip install --no-cache-dir -r requirements.txt
COPY . .
RUN python -m pytest -q
CMD ["python", "run_pipeline.py", "--stage", "all"]
```

الحاوية لا تصلح مشكلة إعادة الإنتاج إذا كانت تعتمد على تثبيتات عائمة. يقرأ Dockerfile ملف requirements.txt المثبت، ويُفضل تسجيل صورة الأساس وبصمة الإصدار. يجب أن يستطيع القارئ بناء الصورة وتشغيل الاختبارات وتنفيذ مرحلة محددة من خط الأنابيب دون تعديل يدوي. إذا فشل البناء بسبب تغير حزمة، فهذه معلومة تشغيلية يجب تسجيلها بدل إخفائها بتثبيت غير معلن.

A container does not guarantee reproducibility if it installs floating dependencies. The Dockerfile should consume a pinned requirements file, and the base image and release fingerprint should be recorded. The reader should be able to build the image, run tests, and execute a defined pipeline stage without manual edits. If a build fails because a dependency changed, record that release information rather than hiding it with an undocumented installation change.

37.9 Docker مع تثبيت الإصدارات

37.9 Docker with Pinned Dependencies

الفصل 38: الاختبار المتكامل — من مثال تعليمي إلى مشروع يمكن مراجعته

Chapter 38: Integrated Verification — From an Educational Example to an Auditable Project

هذا الفصل يضيف طبقة عملية إلى المشروع الحالي. الفكرة ليست زيادة عدد الأدوات، بل جعل المسار من البيانات إلى النتيجة أكثر وضوحًا وقابلية للقياس والمراجعة. كل مثال هنا يستخدم بيانات المشروع أو بيانات تعليمية مصطنعة بوضوح، ولا يعرض نتيجة حسابية على أنها حقيقة تجريبية لم تُختبر.

This chapter adds an operational layer to the existing project. The goal is not to increase the number of tools, but to make the path from data to result clearer, measurable, and reviewable. Each example uses project data or explicitly synthetic educational data, and no computational result is presented as experimental fact unless it has been independently tested.

38.1 اختبار كل طبقة منفردة

38.1 Testing Each Layer Independently

في الاختبار وإعادة الإنتاج لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «اختبار كل طبقة منفردة»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In testing and reproducibility, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Testing Each Layer Independently”, this turns the workflow into an explicit chain of assumptions, data, computations,

and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «اختبار كل طبقة منفردة» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Testing Each Layer Independently” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

اختبار العقود بين الطبقات 38.2

38.2 Testing Contracts Between Layers

في الاختبار وإعادة الإنتاج لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «اختبار العقود بين الطبقات»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In testing and reproducibility, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Testing Contracts Between Layers”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة

«اختبار العقود بين الطبقات» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Testing Contracts Between Layers” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
def assert_close(actual, expected, atol=1e-8):
    import math
    if not math.isclose(actual, expected, abs_tol=atol):
        raise AssertionError((actual, expected))

assert_close(1.000000001, 1.0)
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

38.3 الاختبار المرجعي للنتائج العددية

38.3 Reference Testing for Numerical Results

في الاختبار وإعادة الإنتاج لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «الاختبار المرجعي للنتائج العددية»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In testing and reproducibility, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Reference Testing for Numerical Results”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «الاختبار المرجعي للنتائج العددية» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware.

Therefore, the result of “Reference Testing for Numerical Results” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

هل حددت المدخلات والوحدات والإصدار قبل الحساب؟ ☐

☐ Were the inputs, units, and version defined before computation?

هل يمكن إعادة بناء النتيجة من السجل نفسه؟ ☐

☐ Can the result be reconstructed from the same record?

هل فصلت المثال التعليمي عن الدليل التجريبي؟ ☐

☐ Did you separate the educational example from experimental evidence?

38.4 مقارنة النتائج الكمومية بالمرجع الكلاسيكي

38.4 Comparing Quantum Results With a Classical Reference

في الاختبار وإعادة الإنتاج لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «مقارنة النتائج الكمومية بالمرجع الكلاسيكي»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In testing and reproducibility, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Comparing Quantum Results With a Classical Reference”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو

دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «مقارنة النتائج الكمومية بالمرجع الكلاسيكي» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Comparing Quantum Results With a Classical Reference” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
import hashlib
from pathlib import Path

def sha256(path: Path) -> str:
    h = hashlib.sha256()
    with path.open("rb") as f:
        for chunk in iter(lambda: f.read(1024 * 1024), b""):
            h.update(chunk)
    return h.hexdigest()
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

اختبار البيانات غير المتوقعة 38.5

38.5 Testing Unexpected Data

في الاختبار وإعادة الإنتاج لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «اختبار البيانات غير المتوقعة»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In testing and reproducibility, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Testing Unexpected Data”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column

order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «اختبار البيانات غير المتوقعة» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Testing Unexpected Data” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

38.6 اختبار قابلية إعادة البناء

38.6 Reproducibility Testing

في الاختبار وإعادة الإنتاج لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «اختبار قابلية إعادة البناء»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In testing and reproducibility, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect

it. Applied to “Reproducibility Testing”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «اختبار قابلية إعادة البناء» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Reproducibility Testing” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
def assert_close(actual, expected, atol=1e-8):
    import math
    if not math.isclose(actual, expected, abs_tol=atol):
        raise AssertionError((actual, expected))

assert_close(1.000000001, 1.0)
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

هل حددت المدخلات والوحدات والإصدار قبل الحساب؟ ☐

☐ Were the inputs, units, and version defined before computation?

هل يمكن إعادة بناء النتيجة من السجل نفسه؟ ☐

☐ Can the result be reconstructed from the same record?

هل فصلت المثال التعليمي عن الدليل التجريبي؟ ☐

☐ Did you separate the educational example from experimental evidence?

38.7 تقرير الفشل قبل تقرير النجاح

38.7 Reporting Failure Before Success

في الاختبار وإعادة الإنتاج لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «تقرير الفشل قبل تقرير النجاح»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In testing and reproducibility, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it

produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Reporting Failure Before Success”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

إلى مصفوفة، فيجب تحديد ترتيب SQL عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «تقرير الفشل قبل تقرير النجاح» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Reporting Failure Before Success” must be stated with its limits: what was actually computed, what is an

educational example, and what requires independent data or experiment before it can be treated as evidence.

معيار الجاهزية للنشر 38.8

38.8 Publication-Readiness Criteria

في الاختبار وإعادة الإنتاج لا ينبغي أن يكون الهدف هو إضافة طبقة تقنية لمجرد وجودها. السؤال الأول هو ما المشكلة التي تحلها هذه الخطوة، وما المدخل الذي تحتاجه، وما المخرج الذي تنتجه، وما الدليل الذي يسمح للقارئ أو المراجع بفحصه. عند تطبيق ذلك على «معيار الجاهزية للنشر»، تصبح العملية سلسلة واضحة من افتراضات وبيانات وحسابات ونتائج، بدل أن تكون مجموعة أوامر منفصلة يصعب تفسير العلاقة بينها.

In testing and reproducibility, the goal should not be to add a technical layer merely because it exists. The first questions are what problem the step solves, what input it requires, what output it produces, and what evidence allows a reader or reviewer to inspect it. Applied to “Publication-Readiness Criteria”, this turns the workflow into an explicit chain of assumptions, data, computations, and results rather than a collection of commands whose relationships are difficult to explain.

عمليًا، يجب أن يحمل كل انتقال بين الطبقات عقدًا واضحًا. إذا خرجت البيانات من SQL إلى مصفوفة، فيجب تحديد ترتيب الصفوف والأعمدة، وأنواع البيانات، والوحدات، وطريقة التعامل مع القيم المفقودة. وإذا انتقلت النتيجة إلى نموذج تعلم آلي أو دائرة كمومية، فيجب تسجيل التحويلات والمعاملات والإصدارات. هذا التسجيل ليس عبئًا إضافيًا؛ إنه ما يمنع نتيجة صحيحة حسابيًا من أن تصبح غير قابلة لإعادة البناء.

Operationally, every transition between layers needs an explicit contract. When data move from SQL to a matrix, row and column order, data types, units, and missing-value handling should be defined. When the result moves into a machine-learning model or a quantum circuit, transformations, parameters, and software versions should be recorded. This provenance is not administrative overhead; it is what prevents a computationally correct result from becoming impossible to reconstruct.

والحد العلمي مهم بالقدر نفسه. النموذج قد يصف ارتباطًا ولا يثبت سببًا، والمقترح التوليدي قد يكون صالحًا حسابيًا ولا يكون مادة حقيقية، والمحاكاة الكمومية قد تكون مثالية على محاكي ولا تعكس ضوضاء جهاز حقيقي. لذلك يجب أن تُكتب نتيجة «معيار الجاهزية للنشر» مع حدودها: ماذا تم حسابه بالفعل، وما الذي يمثل مثالًا تعليميًا، وما الذي يحتاج بيانات أو تجربة مستقلة قبل اعتباره دليلًا.

The scientific boundary is equally important. A model may describe an association without establishing causality; a generative proposal may be computationally valid without being a real material; and an ideal quantum simulation may not represent the noise of hardware. Therefore, the result of “Publication-Readiness Criteria” must be stated with its limits: what was actually computed, what is an educational example, and what requires independent data or experiment before it can be treated as evidence.

PYTHON

```
import hashlib
from pathlib import Path

def sha256(path: Path) -> str:
    h = hashlib.sha256()
    with path.open("rb") as f:
        for chunk in iter(lambda: f.read(1024 * 1024), b""):
            h.update(chunk)
    return h.hexdigest()
```

هذا المثال التنفيذي مقصود به إظهار العقد البرمجي بوضوح. قبل استخدامه على بيانات حقيقية يجب تعديل المسارات، والتحقق من الإصدارات، وتشغيل الاختبارات، وتسجيل البيئة والنتيجة. الكود لا يمنح صلاحية علمية تلقائيًا لمخرجاته؛ هو جزء من سلسلة تحقق أكبر.

This executable example is intended to make the software contract explicit. Before using it with real data, paths, versions, tests, environment details, and outputs must be checked and recorded. The code does not automatically make its output scientifically valid; it is one component of a larger verification chain.

التكامل التشغيلي للفصول 27–38

Operational Integration Summary for Chapters 27–38

عند جمع هذه الفصول، تصبح صورة المشروع أكثر تحديدًا: SQL يحفظ الهوية والعلاقات، و NumPy/SciPy يحولان النتائج إلى تمثيلات عددية، وأدوات التعلم الآلي تتعامل مع التنبؤ بعد التحقق من البيانات، والنماذج التوليدية تنتج مرشحين يحتاجون إلى مراجعة، والحوسبة الكمومية تمثل مسارًا متخصصًا يحتاج إلى نمذجة موارد ومقارنة كلاسيكية، بينما تربط الحوكمة كل طبقة بسجل قرار يمكن مراجعته.

Taken together, these chapters make the project architecture more explicit: SQL preserves identity and relationships; NumPy/SciPy provide numerical representations; machine-learning tools handle prediction after data validation; generative models produce candidates that require review; quantum computing is a specialized path that requires resource modeling and classical comparison; and governance connects every layer to a reviewable decision record.

الأهم أن الجدول الدوري بقي مثالًا منظمًا على هندسة البيانات وليس بديلًا عن جينوم كامل. يمكن أن يحمل العدد الذري والرمز والوزن الذري خصائص قابلة للاستعلام والتحليل، ويمكن أن يصبح الهيدروجين مثالًا لمسار VQE، لكن ذلك لا يعني أن المشروع حلل جينومًا بشريًا كاملًا أو أثبت خاصية سريرية. هذه الحدود تحافظ على الأمانة العلمية وتمنع أن يتحول العرض الهندسي إلى ادعاء أكبر من البيانات.

Most importantly, the periodic table remains a structured example for data engineering, not a substitute for a complete genome. Atomic number, symbol, and atomic weight can support query and analysis workflows, and hydrogen can serve as a VQE example, but this does not mean the project analyzed a complete human genome or established a clinical property. These boundaries preserve scientific integrity and prevent an engineering demonstration from becoming a claim larger than its data.

الفشل جزء من البيانات. إذا تعذر تثبيت مكتبة أو فشل استعلام أو لم تتوافر خدمة كمومية، يسجل النظام الحالة والسبب والمرحلة. لا يجوز ملء خانة النتيجة بقيمة افتراضية فقط لكي يكتمل التقرير. القارئ يحتاج أن يعرف أين توقف المسار وما الذي يمكن إعادة تشغيله.

Failure is data. If a dependency cannot be installed, a query fails, or a quantum service is unavailable, the system records the state, reason, and stage. A result field must not be filled with a default value merely to make the report look complete. The reader needs to know where execution stopped and what can be rerun.

عقد الفشل

Failure Contract

كل مخرج يجب أن يحمل اسمًا وإصدارًا وتاريخ تشغيل ومصدرًا. التقرير النصي لا يختلط بملف البيانات، وملف البيانات لا يختلط بنتيجة مشتقة. أما النتيجة الكمومية فتسجل الجهاز أو المحاكي، والدائرة، وعدد القياسات، والإعدادات الأساسية. هذا الفصل بين المدخل والمخرج يمنع الالتباس عند إعادة التشغيل بعد أشهر.

Every output should carry a name, version, run date, and source. The narrative report should not be confused with raw data, and raw data should not be confused with derived results. A quantum result should record the simulator or backend, circuit, shot count, and key settings. This separation prevents ambiguity when the workflow is rerun months later.

عقد الإخراج

Output Contract

هذا الملحق لا يضيف أداة جديدة؛ بل يحول ما سبق إلى عقد تشغيل واحد. النسخة القابلة للنشر يجب أن تحتوي على مصدر بيانات معروف، ونسخة بيئة محددة، واختبارات ناجحة، وسجل تشغيل، ومخرجات يمكن ربطها بإصدار الكود. إذا تعذر تشغيل مرحلة اختيارية مثل العتاد الكمومي، يسجل ذلك صراحةً ولا يعاد تعريف الفشل على أنه نجاح.

This appendix does not add another tool; it turns the preceding material into a single execution contract. A releasable project needs a known data source, a defined environment, passing tests, a run record, and outputs linked to a code version. If an optional stage such as real quantum hardware cannot run, that limitation is recorded explicitly rather than relabeled as success.

Operational Labs: From Concept to Test

إذا فشل تحميل مكتبة أو انتهت مهلة خدمة كمومية أو فشل اختبار، يجب أن ينتج التشغيل سجلًا يحتوي على run_id وstage وstatus وerror_type وmessage وtimestamp. هذا يمنع إعادة تشغيل المشروع عشوائيًا بحثًا عن نجاح عابر، ويجعل الفشل نفسه مادة يمكن تحليلها وتحسين النظام بناءً عليها.

If a dependency installation fails, a quantum service times out, or a test fails, the run should produce a record containing run_id, stage, status, error_type, message, and timestamp. This prevents random reruns in search of a transient success and turns failure itself into evidence that can be analyzed to improve the system.

دراسة حالة 10: فشل التشغيل نتيجة يمكن تتبعها

Case Study 10: An Execution Failure Is a Traceable Result

وظيفة Gene Resistance Tracker في المشروع هي تنظيم الأدلة وربطها بالإصدارات ومصادرها. إذا سجل متغيرًا مرتبطًا بالمقاومة في دراسة، فهذا لا يعني أن النظام أثبت مقاومة فرد بعينه. يجب أن يبقى الفرق بين evidence record وclinical interpretation واضحًا في أسماء الجداول والحقول والتقارير.

The role of the Gene Resistance Tracker is to organize evidence and link it to versions and sources. If a study reports a resistance-associated variant, the system has not thereby proven resistance in a particular individual. The distinction between an

evidence record and a clinical interpretation should remain explicit in tables, fields, and reports.

دراسة حالة 9: متتبع المقاومة لا يصدر حكمًا طبيًا

Case Study 9: The Resistance Tracker Does Not Issue a Medical Judgment

عند عرض دائرة من 4–6 كيوبت، يجب أن يوضح الرسم ما الذي تم تمثيله وما الذي لم يتم تمثيله. قد تكون الدائرة مثالًا على mapping أو Hamiltonian مبسط أو حالة متعددة الإلكترونات، لكنها لا تعني أن جميع المدارات الإلكترونية لجزيء H_2O أو CO_2 موجودة فيها. هذه العبارة ليست تحفظًا لغويًا فقط؛ إنها جزء من تعريف التجربة.

When presenting a 4–6-qubit circuit, the figure should state what is represented and what is not. The circuit may illustrate a mapping, a reduced Hamiltonian, or a multi-electron example, but it does not imply that all electronic orbitals of H_2O or CO_2 are present. This is not merely a wording precaution; it is part of the experiment definition.

دراسة حالة 8: 4–6 كيوبت مثال على البنية لا على اكتمال الجزيء

Case Study 8: 4–6 Qubits Demonstrate Architecture, Not Molecular Completeness

يجب أن يوضح تقرير VQE ما إذا كانت الطاقة ناتجة عن محاكي مثالي أو محاكي ضوضائي أو عتاد حقيقي. ثم يذكر مرجعًا كلاسيكيًا، وعدد القياسات، وإعدادات المحسن، وأي تقنية لتخفيف الخطأ. يمكن تقسيم الفرق النهائي إلى مصادر: خطأ النموذج، خطأ التقريب، خطأ الخوارزمية، وضوضاء العتاد عندما تكون ذات صلة.

A VQE report should state whether the energy came from an ideal simulator, noisy simulator, or real hardware. It should include a classical reference, shot count, optimizer settings, and any error-mitigation method. The final discrepancy can then be

discussed in terms of model error, approximation error, algorithmic error, and hardware noise when applicable.

دراسة حالة 7: المحاكاة الكمومية تحتاج مرجعًا وميزانية خطأ

Case Study 7: Quantum Simulation Needs a Reference and an Error Budget

إذا أنتج GAN أو VAE مرشحًا جديدًا، فالمرحلة التالية ليست نشره في التقرير بوصفه اكتشافًا. يجب التحقق من الصلاحية البنيوية، ثم حساب خواص مستقلة، ثم مقارنة المرشح بنطاق بيانات التدريب، ثم تحديد ما إذا كانت هناك طريقة تجريبية معقولة لاختباره. في المشروع التعليمي يمكن الاكتفاء بالخطوات الحسابية مع التصريح بأنها فرضية قابلة للاختبار.

When a GAN or VAE generates a new candidate, the next step is not to publish it as a discovery. Check structural validity, calculate independent properties, compare the candidate with the training domain, and determine whether a plausible experimental route exists. In the educational project, the computational checks may be demonstrated while clearly labeling the output as a testable hypothesis.

دراسة حالة 6: التوليد لا يلغي التحقق الفيزيائي

Case Study 6: Generation Does Not Replace Physical Validation

اقتراح مادة أو خليط عناصر قد يتضمن أكثر من هدف: خواص ميكانيكية، توصيل كهربائي، كثافة، كلفة، أو قيود بيئية. لا توجد عادةً إجابة واحدة مثالية. يمكن بناء Pareto front ثم عرض المرشحين مع أسباب ترتيبهم. إذا تغيرت الأوزان، يجب أن تتغير التوصية بطريقة يمكن تفسيرها، ويجب حفظ دالة الهدف مع كل تشغيل.

A material or element-mixture recommendation may involve multiple objectives: mechanical properties, electrical conductivity, density, cost, or environmental constraints. There is often no single optimum. A Pareto front can expose candidates and their trade-offs.

If objective weights change, the recommendation should change in an explainable way, and the objective function must be stored with each run.

دراسة حالة 5: التوصية متعددة الأهداف

Case Study 5: Multi-Objective Recommendation

إذا تنبأ النموذج بخاصية لعنصر لم توجد له قيمة مرجعية، يجب أن تسمى النتيجة prediction لا measurement. يجب أن يظهر نطاق التدريب ومقياس الخطأ ودرجة عدم اليقين، وأن يعرف القارئ إن كان العنصر خارج نطاق الخصائص التي تعلم منها النموذج. هذه التفرقة ضرورية عندما تتحول النتيجة إلى اقتراح بحثي.

If a model predicts a property for an element with no reference value, the result should be labeled a prediction, not a measurement. Report the training domain, error metric, and uncertainty, and state whether the element is outside the feature distribution used for training. This distinction matters when a prediction becomes a research proposal.

دراسة حالة 4: نموذج يتنبأ بخاصية لا يعني أنه قاسها

Case Study 4: A Prediction Is Not a Measurement

المصفوفة العددية لا تحمل عادةً اسم العنصر داخل كل خلية. لذلك يجب حفظ جدول مرافق يربط رقم الصف بالعدد الذري والرمز. عند إعادة ترتيب الصفوف أو حذف خاصية، لا يكفي حفظ ملف NPY وحده؛ يجب حفظ feature_names وelement_identifiers وdataset_version. هذا يمنع خطأ شائعًا: تحليل مصفوفة صحيح عدديًا لكن تفسير الصفوف فيها خاطئ.

A numerical matrix usually does not carry the element identity inside every cell. Store a companion mapping from row position to atomic number and symbol. When rows are reordered or features removed, an NPY file alone is insufficient; preserve feature_names, element_identifiers, and dataset_version. This prevents a common

failure: numerically correct analysis with incorrect row interpretation.

دراسة حالة 3: تحويل SQL إلى مصفوفة دون فقد الهوية

Case Study 3: Moving from SQL to a Matrix Without Losing Identity

عندما تظهر قيمتان مختلفتان لخاصية واحدة، لا ينبغي اختيار إحداهما يدويًا ثم حذف الأخرى. يسجل النظام المصدر، والإصدار، والوحدة، وتاريخ الاسترجاع، ثم يحدد قاعدة اختيار معلنة. إذا كانت القيمة مشتقة من قيمة أخرى، يجب أن يسجل ذلك في lineage بدل أن تظهر كأنها قياس مستقل.

When two values disagree for the same property, the system should not silently choose one and discard the other. It should record source, version, unit, retrieval date, and an explicit selection rule. If a value is derived from another value, that lineage should be stored instead of presenting it as an independent measurement.

دراسة حالة 2: مصدر القيمة جزء من البيانات

Case Study 2: Value Provenance Is Part of the Data

إذا احتاج المشروع إلى تمثيل النظائر، لا نضع قائمة النظائر داخل عمود نصي في elements. ننشئ جدولًا مستقلًا يربط كل نظير بالعدد الذري، ويخزن mass_number وisotope_mass وabundance وsource_reference. بهذه الطريقة يمكن إضافة نظير جديد دون تغيير هوية العنصر الأساسي، ويمكن إجراء استعلامات تجمع بين العنصر والنظير مع بقاء المفاتيح واضحة.

If the project needs isotope representation, the isotopes should not be stored as a text list inside elements. A separate table can link each isotope to atomic_number and store mass_number, isotope_mass, abundance, and source_reference. A new isotope can then be added without changing the identity of the base element, while joins between elements and isotopes remain explicit.

دراسة حالة 1: إضافة النظائر دون كسر المخطط

Case Study 1: Adding Isotopes Without Breaking the Schema

الفصل التاسع والثلاثون: التطبيق المتكامل الصغير — SQL وNumPy وscikit-learn

Chapter 39: A Small Integrated Application — SQL, NumPy, and scikit-learn

39.1 لماذا نحتاج مثالًا صغيرًا كامل المسار؟

39.1 Why a Small End-to-End Example Matters

الفصل يضع مسارًا يمكن تشغيله من البداية إلى النهاية على جهاز واحد: إنشاء قاعدة SQLite، إدخال سجل عناصر مضبوط، استخراج مصفوفة عددية، تقسيم البيانات، تدريب نموذج بسيط، ثم حفظ نتيجة مرتبطة بإصدار البيانات. قيمة المثال ليست في دقة تنبؤ خارقة، بل في إظهار العقد بين الطبقات.

This chapter defines a workflow that can run end to end on one machine: create SQLite storage, load controlled element records, extract a numerical matrix, split the data, train a small model, and persist a result with a dataset version. Its value is not an impressive score; it is the explicit contract between layers.

عند فشل أي مرحلة يجب أن يتوقف المسار بدل إنتاج تقرير يبدو كاملًا. فشل الاتصال بقاعدة البيانات، أو وجود عمود ناقص، أو اختلاف عدد الصفوف بين المعرّفات والمصفوفة، كلها حالات يجب اكتشافها قبل التدريب. هذا هو الفرق بين مثال برمجي ومشروع يمكن مراجعته.

If any stage fails, the pipeline should stop rather than produce a report that merely looks complete. A missing database column, a row-count mismatch, or an identifier/matrix mismatch must be

detected before training. That is the difference between a code sample and an auditable project.

39.2 عقد البيانات بين SQL وNumPy

39.2 The SQL-to-NumPy Data Contract

يجب تثبيت ترتيب الأعمدة وتوثيق الوحدات والقيم المفقودة. لا يكفي أن NumPy قبل التحويل إلى تكون المصفوفة ذات شكل صحيح؛ قد تكون كل الأبعاد صحيحة بينما وُضع الوزن الذري في عمود الكثافة أو العكس. لذلك يحمل العقد أسماء الأعمدة وترتيبها وإصدار البيانات.

Before converting to NumPy, column order, units, and missing-value policy must be fixed. A matrix can have the correct shape while still putting atomic weight in the density column. The contract therefore records column names, order, units, and dataset version.

التحويل الآمن يترك معرّف العنصر خارج المصفوفة العددية أو في بنية منفصلة. عند إعادة بناء النتائج يجب أن يعود كل صف إلى العدد الذري والرمز الأصليين. هذا يمنع خطأً صامتًا قد يجعل النموذج يتنبأ بقيمة صحيحة حسابيًا ولكن ينسبها إلى عنصر آخر.

A safe transformation keeps element identity outside the numerical matrix or in a separate structure. When results are reconstructed, every row must map back to its original atomic number and symbol. This prevents a silent error in which a model predicts a numerically valid value for the wrong element.

39.3 اختيار خط أساس بسيط

39.3 Choosing a Simple Baseline

صغير Random Forest الخط الأساس ليس مرحلة بدائية يجب التخلص منها. نموذج خطي أو يعطي نقطة مرجعية يمكن مقارنة أي نموذج أعمق بها. إذا لم يتفوق النموذج المعقد بعد ضبط التقييم، فلا يوجد سبب تشغيلي لإضافته إلى النظام.

A baseline is not a primitive stage to discard. A small linear model or Random Forest provides a reference against which deeper models can be compared. If the complex model does not improve the validated evaluation, there is no operational reason to add it.

في البيانات الصغيرة يجب أن يكون الفصل بين التدريب والاختبار واضحًا، وأن يكون اختيار الميزات داخل عملية التقييم. إذا اختيرت الميزات باستخدام جميع الصفوف قبل التقسيم، فقد يتسرب جزء من معلومات الاختبار إلى التدريب. لذلك يسجل التقرير طريقة التقسيم والبذرة العشوائية ومقياس التقييم.

With small data, train/test separation must be explicit and feature selection must occur inside the evaluation procedure. Selecting features from all rows before splitting can leak information from the test set into training. The report therefore records the split method, random seed, and evaluation metric.

39.4 حفظ النتيجة كسجل تجربة

39.4 Persisting the Result as an Experiment Record

لا نحفظ رقم الدقة وحده. نحفظ اسم النموذج، إصدار البيانات، إصدار الكود، عدد الصفوف، المقياس، زمن التنفيذ، ومسار المخرج. بهذه المعلومات يستطيع قارئ آخر معرفة أي تشغيل أنتج الرقم حتى لو تغيرت البيئة لاحقًا.

Do not persist only an accuracy value. Store the model name, dataset version, code version, row count, metric, execution time, and output path. This context lets another reader identify which run produced the number even after the environment changes.

النتيجة التي لا تحمل سياقها تصبح رقمًا معزولًا. أما النتيجة المسجلة مع سياقها فيمكن فحصها وإعادة إنتاجها ومقارنتها بتشغيل لاحق. هذه الفكرة هي التطبيق العملي لمفهوم provenance الذي يمتد من SQL إلى النماذج والكم.

A result without context becomes an isolated number. A result stored with context can be inspected, reproduced, and compared

with a later run. This is the practical meaning of provenance across SQL, machine learning, and quantum workflows.

39.5 كود التطبيق المتكامل

39.5 Integrated Application Code

الكود التالي مكتفٍ بذاته لأغراض الاختبار المحلي. يستخدم بيانات صغيرة مصطنعة حتى لا يخلط المثال بين اختبار البنية وبين ادعاء وجود مجموعة تدريب علمية كبيرة. يمكن استبدال طبقة الإدخال ببيانات المشروع الموثقة مع الحفاظ على العقد نفسه.

The following code is self-contained for local testing. It uses a small synthetic dataset so that structural testing is not confused with a claim of a large scientific training corpus. The input layer can later be replaced with documented project data while keeping the same contract.

بعد تشغيل الكود يجب أن ترى عدد الصفوف، شكل المصفوفة، ومقياس النموذج. إذا تغير ترتيب الأعمدة أو عدد الصفوف فسيوقف الكود التنفيذ. هذا الفشل المقصود أفضل من نتيجة تبدو صحيحة ولا يمكن الوثوق في نسبتها.

After execution, the output should show row count, matrix shape, and a model metric. If column order or row count changes, the program stops. This deliberate failure is safer than producing a plausible but untrustworthy result.

PYTHON / integrated_sql_numpy_sklearn.py

```
from __future__ import annotations
import sqlite3
import numpy as np
from sklearn.ensemble import RandomForestRegressor

DB = ":memory:"
rows = [
    (1, "H", 1.008, 0.0899),
    (2, "He", 4.003, 0.1785),
    (6, "C", 12.011, 2.267),
    (8, "O", 15.999, 1.429),
    (14, "Si", 28.085, 2.329),
    (26, "Fe", 55.845, 7.874),
]
with sqlite3.connect(DB) as con:
    con.execute("CREATE TABLE elements (atomic_number INTEGER PRIMARY KEY, symbol TEXT,
```

```
atomic_weight REAL, density REAL)")
    con.executemany("INSERT INTO elements VALUES (?, ?, ?, ?)", rows)
    data = con.execute("SELECT atomic_number, atomic_weight, density FROM elements ORDER BY
atomic_number").fetchall()

X = np.asarray([[r[0], r[1]] for r in data], dtype=float)
y = np.asarray([r[2] for r in data], dtype=float)
assert X.shape == (6, 2)
assert np.isfinite(X).all() and np.isfinite(y).all()
model = RandomForestRegressor(n_estimators=32, random_state=7).fit(X, y)
pred = model.predict(X)
print({"rows": len(data), "shape": X.shape, "mae": float(np.mean(np.abs(pred - y)))})
```

الفصل الأربعون: بيانات خواص المواد — من المصدر الحقيقي إلى مجموعة تدريب قابلة للتتبع

Chapter 40: Materials-Property Data — From a Real Source to a Traceable Training Set

40.1 لماذا نحتاج بيانات تتجاوز العناصر؟

40.1 Why We Need Data Beyond Individual Elements

العنصر الواحد لا يمثل مادة. عندما ننتقل إلى خواص المواد تظهر البنية الكيميائية، التركيب، الكثافة، فجوة الطاقة، طاقة التكوين، والظروف التي قيس فيها كل متغير. لذلك لا يجوز أن نعامل جدول العناصر كأنه مجموعة خواص مواد جاهزة للتعلم العميق.

An element is not a material. Material-property data introduce composition, structure, density, band gap, formation energy, and the conditions under which each quantity was measured or calculated. The periodic table must therefore not be treated as a ready-made materials dataset for deep learning.

المسار العملي يبدأ بمصدر بيانات حقيقي موثق مثل قاعدة بيانات مواد مرخصة أو مجموعة منشورة. نحتفظ باسم المصدر، تاريخ الاسترجاع، الترخيص، رقم الإصدار أو الاستعلام، والوحدات. إذا كانت البيانات محسوبة لا مقاسة، يجب تسجيل المنهج الحسابي أيضًا.

A practical workflow starts from a documented real source such as a licensed materials database or a published dataset. Store the source name, retrieval date, license, release or query identifier, and units. If values are computed rather than measured, record the computational method as well.

40.2 مخطط سجل المادة

40.2 Material-Record Schema

السجل الجيد يفصل الهوية عن الخواص. يمكن أن يحتوي material_id وformula وstructure_id وsource_reference وdataset_version، ثم جدولًا للخواص يضم property_name وvalue وunit وmethod. هذا التصميم يمنع خلط قيم من مصادر أو طرق مختلفة داخل عمود واحد.

A robust record separates identity from properties. A material table can contain material_id, formula, structure_id, source_reference, and dataset_version, while a property table stores property_name, value, unit, and method. This prevents values from different sources or methods from being silently mixed.

وجود unit وmethod ليس ترفًا. قيمة الكثافة مثلًا لا يمكن تفسيرها دون معرفة وحدتها، كما أن فجوة الطاقة قد تختلف باختلاف المنهج الحسابي. عندما تدخل القيم إلى نموذج تعلم آلي يجب أن تبقى هذه المعلومات مرتبطة بالصف الذي أنتج الميزة.

Unit and method are not optional decoration. A density value cannot be interpreted without its unit, and band gaps can differ by computational method. When values enter a machine-learning model, this context should remain linked to the row from which the feature was derived.

40.3 فصل البيانات قبل التدريب

40.3 Splitting Data Before Training

في بيانات المواد قد تكون الصيغة نفسها أو عائلة تركيبية متقاربة موجودة في التدريب والاختبار. عندها قد يبدو النموذج قويًا لأنه رأى أمثلة شديدة التشابه. لذلك قد يكون GroupKFold أو تقسيم حسب عائلة المواد أكثر صدقًا من تقسيم عشوائي بسيط.

Materials datasets can contain the same formula family or closely related structures in both training and test sets. A model may then appear strong because it has seen near-duplicates. GroupKFold or family-based splitting can therefore be more honest than a simple random split.

يجب أيضًا فصل البيانات قبل حساب أي إحصاء يستخدم في التحويل، مثل متوسطات التعويض أو معاملات التقييس. إذا حُسبت هذه القيم من المجموعة كاملة ثم قُسمت، دخلت معلومات الاختبار إلى التدريب بصورة غير مباشرة.

Data should also be split before computing transformation statistics such as imputation means or scaling parameters. Computing them on the full dataset before splitting leaks test information into training indirectly.

40.4 توثيق البيانات الحقيقية

40.4 Documenting Real-World Data

المشروع لا يضيف أرقامًا خارج مصدرها لإكمال جدول. عندما لا تتوفر قيمة موثقة نتركها مفقودة أو نضعها في طبقة بيانات منفصلة. الرقم المختلق أخطر من الخانة الفارغة لأنه قد يمر إلى النموذج ويبدو كأنه حقيقة.

The project must not add undocumented numbers merely to complete a table. If a value is not available from a traceable source, keep it missing or place it in a separate modeled-data layer. A

fabricated number is more dangerous than an empty field because it can silently enter a model as if it were fact.

وألا synthetic=true للتجارب التعليمية يمكن استخدام بيانات مصطنعة، لكن يجب أن تحمل علامة تختلط ببيانات المصدر الحقيقي. هذا الفصل بين الطبقات يسمح للقارئ بتشغيل الكود الآن ثم استبدال المدخلات لاحقًا دون إعادة كتابة منطق التحليل.

Synthetic data are appropriate for teaching, but they should carry a synthetic=true marker and must not be mixed with source data. This separation lets the reader run the code now and later replace inputs without rewriting the analysis logic.

40.5 محول CSV قابل للاختبار

40.5 A Testable CSV Adapter

المحول التالي لا يفترض مصدرًا بعينه؛ إنه يفرض عقد الأعمدة والوحدات على ملف خارجي. بهذه الطريقة يصبح إدخال بيانات حقيقية خطوة منفصلة يمكن مراجعتها قبل وصول البيانات إلى النموذج.

The following adapter does not assume a particular provider. It enforces a column contract and basic numeric checks on an external file. This makes real-data ingestion a separate reviewable stage before the model sees the values.

إذا كانت البيانات تأتي من واجهة API، تحفظ الاستجابة الخام أو بصمتها، ثم تُحول إلى نفس المخطط. بذلك يمكن الرجوع إلى المصدر الأصلي عند ظهور اختلاف في نتيجة النموذج أو عند تحديث مجموعة البيانات.

If data arrive through an API, preserve the raw response or its hash and then transform it into the same schema. This makes it possible to return to the source when a model result changes or the dataset is updated.

PYTHON / materials_csv_adapter.py

```
from __future__ import annotations
from pathlib import Path
import pandas as pd
```

```
REQUIRED = {"material_id", "formula", "property_name", "value", "unit", "source_reference",
"dataset_version"}

def load_material_properties(path: str) -> pd.DataFrame:
    df = pd.read_csv(Path(path))
    missing = REQUIRED - set(df.columns)
    if missing:
        raise ValueError(f"Missing required columns: {sorted(missing)}")
    if df["material_id"].isna().any() or df["source_reference"].isna().any():
        raise ValueError("Identity and provenance fields cannot be null")
    df["value"] = pd.to_numeric(df["value"], errors="raise")
    return df

if __name__ == "__main__":
    print("Adapter ready: provide a documented CSV source to run the ingestion step.")
```

الفصل الحادي والأربعون: Jupyter وColab وBinder — العرض التفاعلي دون فقدان قابلية الإعادة

Chapter 41: Jupyter, Colab, and Binder — Interactive Work Without Losing Reproducibility

41.1 الدفتر ليس بديلًا عن المصدر

41.1 A Notebook Is Not the Source of Truth

الدفتر التفاعلي ممتاز لشرح تجربة خطوة بخطوة، لكنه يسمح أيضًا بتشغيل الخلايا بترتيب غير خطي. لذلك يجب أن تكون ملفات Python وSQL هي المصدر التشغيلي، بينما يستخدم الدفتر لعرض النتائج والرسوم وقرارات التحليل.

An interactive notebook is excellent for explaining an experiment step by step, but it also allows cells to be executed out of order. Python and SQL files should therefore remain the operational source of truth, while the notebook presents results, plots, and analysis decisions.

إذا فشلت خلية لأن متغيرًا أنشئ في جلسة قديمة، فهذه إشارة إلى مشكلة في قابلية الإعادة وليست مجرد إزعاج في الدفتر. قبل النشر يجب إعادة تشغيل الدفتر من نواة جديدة باستخدام Run All.

Before release, restart the kernel and run all cells. If a cell fails because a variable existed only in an earlier session, that is a reproducibility defect rather than a minor notebook inconvenience.

41.2 بنية دفتر المشروع

41.2 Notebook Structure

البنية المقترحة تبدأ ببيانات البيئة، ثم تحميل البيانات، ثم التحقق، ثم التحويل، ثم النموذج، ثم التقييم، ثم حفظ النتائج. كل قسم يعرض مدخلاته ومخرجاته ولا يعتمد على أثر جانبي غير موثق في خلية بعيدة.

A robust notebook begins with environment information, then data loading, validation, transformation, modeling, evaluation, and result persistence. Each section exposes its inputs and outputs instead of depending on undocumented side effects from a distant cell.

يجب أن يظهر إصدار البيانات والبرنامج في أول الدفتر. وإذا كان هناك رقم عشوائي في التعلم الآلي أو المحاكاة، تسجل البذرة. وإذا كانت النتيجة تعتمد على خدمة خارجية، يسجل اسم الخدمة والواجهة والإصدار وحالة المصادقة دون وضع الأسرار داخل الدفتر.

The dataset and software versions should appear near the beginning of the notebook. If randomness is used, record the seed. If a result depends on an external service, record the service, interface, version, and authentication state without embedding secrets in the notebook.

41.3 Colab وBinder كواجهات تشغيل

41.3 Colab and Binder as Execution Interfaces

Colab أو Binder يمكن لبيئة سحابية أن تخفض عتبة التشغيل، لكنها لا تلغي الحاجة إلى ملف بيئة ثابت. رابط Binder يجب أن يشير إلى مستودع محدد الإصدار، وأن يقرأ requirements أو environment من المصدر بدل الاعتماد على بيئة مجهولة.

A hosted environment can lower the barrier to execution, but it does not remove the need for a pinned environment. A Colab or Binder link should point to a versioned repository and install dependencies from the project files rather than relying on an unknown ambient environment.

إذا تغيرت مكتبة بعد ستة أشهر، قد تتغير النتيجة أو يتوقف الكود. لذلك تحفظ نسخة الإصدار وتاريخ الاختبار. الهدف ليس تجميد العلم، بل جعل التغيير ظاهرًا وقابلًا للتشخيص.

If a library changes six months later, results may change or code may fail. Record the tested version and date. The goal is not to freeze science; it is to make change visible and diagnosable.

41.4 تحويل الدفتر إلى اختبار

41.4 Turning Notebook Claims into Tests

كل ادعاء قابل للقياس يمكن تحويله إلى اختبار. إذا قال الدفتر إن عدد العناصر 118، يصبح ذلك assertion. وإذا قال إن المصفوفة لا تحتوي NaN بعد سياسة التعويض، يصبح ذلك اختبارًا. وإذا قال إن الطاقة المرجعية تقع ضمن هامش، يصبح اختبارًا عدديًا.

Every measurable notebook claim can become a test. If the notebook says there are 118 elements, assert it. If it says the matrix has no NaNs after the imputation policy, test it. If it gives an energy reference within a tolerance, encode that as a numerical test.

بهذه الطريقة لا يبقى الدفتر عرضًا منفصلًا عن المشروع. يصبح واجهة تعليمية لنتائج يمكن اختبارها في CI، وتبقى الخلايا قابلة لإعادة التشغيل دون أن تصبح المصدر الوحيد للمنطق.

The notebook then becomes an educational interface over results that can also be tested in CI. Cells remain useful for exploration without becoming the only place where the logic exists.

41.5 سجل تشغيل الدفتر

41.5 Notebook Execution Record

يُحفظ run_id وdataset_version وenvironment_hash وtimestamp مع كل تشغيل مهم. وإذا تغيرت الخلايا، يمكن حفظ نسخة الدفتر أو بصمتها. البصمة لا تثبت صحة المحتوى، لكنها تربط التقرير بالملف الذي شغّل التجربة.

Store run_id, dataset_version, environment_hash, and timestamp for important notebook executions. If cells change, preserve the notebook or its hash. A hash does not prove scientific correctness, but it links the report to the file that produced the run.

هذا ينسجم مع مبدأ أن النتيجة ليست رقمًا فقط. النتيجة هي رقم مع مصدره وطريقة حسابه وإصداره وسجل تشغيله. كلما اقترب النظام من بيانات حقيقية، تصبح هذه التفاصيل أكثر أهمية لا أقل.

This follows the rule that a result is not merely a number. It is a number with its source, computation method, version, and execution record. As the system approaches real data, these details become more important, not less.

PYTHON / notebook_manifest.py

```
from __future__ import annotations
import hashlib
import json
from pathlib import Path

def notebook_manifest(notebook_path: str, dataset_version: str, environment_version: str) -> dict:
    path = Path(notebook_path)
    digest = hashlib.sha256(path.read_bytes()).hexdigest()
    return {
        "notebook": path.name,
        "notebook_sha256": digest,
        "dataset_version": dataset_version,
        "environment_version": environment_version,
    }

if __name__ == "__main__":
    manifest = notebook_manifest("analysis.ipynb", "elements-v1", "python-3.12")
    Path("notebook_manifest.json").write_text(json.dumps(manifest, indent=2), encoding="utf-8")
    print(manifest)
```

الفصل الثاني والأربعون: HHL في نظام خطي صغير — التنفيذ، المرجع الكلاسيكي وحدود الادعاء

Chapter 42: HHL on a Small Linear System — Execution, Classical Baseline, and Claim Boundaries

42.1 ما الذي تحله HHL فعلًا؟

42.1 What HHL Actually Solves

HHL مخصصة لمسائل خطية من الشكل A x = b تحت افتراضات محددة حول تمثيل A، وحالة b، وعدد الشروط، ودقة الطور، واستخراج المعلومة المطلوبة من الحالة الكمومية. لذلك لا يصح وصفها بأنها خوارزمية عامة لحل معادلة شرودنجر مباشرة.

HHL targets linear systems of the form A x = b under assumptions about the representation of A, the state |b>, conditioning, phase precision, and the information extracted from the quantum state. It should not be described as a general direct solver for the Schrödinger equation.

التنفيذ التعليمي هنا يستخدم مصفوفة صغيرة قابلة للحل بدقة، ثم يقارن الحل المرجعي مع تمثيل تطبيع للمتجه. هذا يجعل المقارنة قابلة للحساب دون ادعاء أن المحاكاة البرمجية تعادل سرعة جهاز كمومي حقيقي.

The educational execution uses a small exactly solvable matrix and compares the classical reference with a normalized vector representation. This makes the comparison computable without claiming that a software simulation provides the speed of real quantum hardware.

42.2 المرجع الكلاسيكي قبل الكم

42.2 The Classical Reference Before Quantum Execution

قبل أي تشغيل كمومي يجب حل النظام كلاسيكيًا. نسجل x_ref، ونحسب معيار الخطأ عندما يظهر حل مرشح، ونحفظ condition number للمصفوفة. إذا كانت المصفوفة سيئة التكييف، فقد يصبح الخطأ حساسًا حتى لو كانت الدائرة الكمومية نفسها صحيحة.

Before any quantum execution, solve the system classically. Store x_ref, compute an error norm when a candidate solution is available, and record the matrix condition number. An ill-conditioned matrix can amplify error even when the quantum circuit is implemented correctly.

المرجع الكلاسيكي لا ينافس الكم بحد ذاته؛ وظيفته حماية التفسير. إذا ظهرت قيمة مختلفة، لا نعرف أنها تقدم كمومي إلا بعد فصل أخطاء التمثيل والضوضاء والدقة والتحضير عن جودة الخوارزمية.

The classical reference is not a competition with quantum computing. Its role is interpretive protection. A different value is not evidence of a quantum advantage until representation, noise, precision, state preparation, and algorithmic effects are separated.

42.3 تمثيل صغير قابل للاختبار

42.3 A Small Testable Representation

نستخدم A = diag(1,3) و b = [1,1]. الحل الكلاسيكي هو [1,1/3]. بعد التطبيع نحصل على حالة متجهية يمكن مقارنتها بخرج محاكاة HHL التعليمية. المثال صغير عمدًا لأن الهدف هو فهم المراحل، لا ادعاء مقياس عملي واسع.

Use A = diag(1,3) and b = [1,1]. The classical solution is [1,1/3]. After normalization, the vector becomes a state representation that can be compared with an educational HHL simulation. The example is intentionally small because the goal is to expose the stages, not to claim large-scale utility.

وعدد كيوبتات سجل الحل، ودقة الطور، وطريقة القياس، وأي condition number، يجب تسجيل post-selection. إذا لم نسجل هذه العناصر فلن يمكن تفسير اختلاف النتيجة بين تشغيلين.

Record the condition number, solution-register qubit count, phase precision, measurement method, and any post-selection. Without these fields, differences between runs cannot be interpreted reliably.

42.4 محاكاة HHL العددية

42.4 Numerical HHL Demonstrator

الكود التالي يحقق الجزء القابل للتحقق عدديًا: المرجع، التطبيع، condition number، وخطأ المقارنة. أما الدائرة الكمومية الكاملة فتحتاج بيئة Qiskit متوافقة، وتبقى منفصلة عن هذا الاختبار حتى لا تختلط صحة الرياضيات بصحة واجهة المكتبة.

The following code executes the numerically verifiable part: reference solution, normalization, condition number, and comparison error. A full quantum circuit requires a compatible Qiskit environment and remains a separate test so mathematical correctness is not confused with library-interface correctness.

عندما يكون التنفيذ عبر الدائرة الكمومية الفعلية يجب استخدام V2 primitives عند تشغيل Qiskit Runtime، مع تسجيل backend وshots وtranspilation والإصدار. الواجهة الحالية لـEstimatorV2 وSamplerV2 تعتمد على PUBs وتفصل بين الدائرة التي صارت ISA وبين عملية التنفيذ.

When running on actual quantum services, use V2 primitives for Qiskit Runtime and record backend, shots, transpilation, and software version. The current EstimatorV2 and SamplerV2 interfaces use PUBs and distinguish ISA circuits from the execution step.

42.5 متى تصبح HHL غير مناسبة؟

42.5 When HHL Is the Wrong Tool

إذا كانت المسألة صغيرة جدًا، فإن حلها مباشرة بكود عددي قد يكون أبسط وأسرع. وإذا كانت عملية تحميل البيانات إلى الحالة الكمومية مكلفة، فقد تلتهم هذه الكلفة أي فائدة نظرية. وإذا كانت المعلومة المطلوبة قيمة واحدة يمكن حسابها كلاسيكيًا، فلا ينبغي تحويل المسألة إلى كم لمجرد استخدام التقنية.

For very small systems, direct numerical linear algebra may be simpler and faster. If state preparation is expensive, it can dominate any theoretical benefit. If the required output is a scalar that is easy to compute classically, the problem should not be moved to quantum hardware merely to use the technology.

هذه ليست قيودًا على قيمة الحوسبة الكمومية؛ إنها قواعد اختيار أداة. المشروع يعلم القارئ أن يختار الخوارزمية عندما تضيف قدرة قابلة للقياس، وأن يسجل لماذا كانت الطريقة المختارة مناسبة للمسألة.

These are not arguments against quantum computing; they are tool-selection rules. The project teaches the reader to choose an algorithm when it adds measurable capability and to record why the chosen method fits the problem.

PYTHON / hhl_small_system_reference.py

```
from __future__ import annotations
import numpy as np

A = np.diag([1.0, 3.0])
b = np.array([1.0, 1.0])
x_ref = np.linalg.solve(A, b)
condition = float(np.linalg.cond(A))
x_state = x_ref / np.linalg.norm(x_ref)

# A normalized candidate state is the object compared with an HHL-style simulation.
assert np.isclose(np.linalg.norm(x_state), 1.0)
print({
    "x_ref": x_ref.tolist(),
    "normalized_solution": x_state.tolist(),
    "condition_number": condition,
})
```

PYTHON / quantum_environment_smoke_test.py

```
from qiskit import QuantumCircuit

# Minimal executable circuit used to verify the quantum runtime environment.
# It does not claim to implement general HHL; it checks the circuit layer before
# a separately validated HHL implementation is attached to the experiment.
```

```
qc = QuantumCircuit(2)
qc.h(0)
qc.cx(0, 1)
qc.measure_all()
print(qc)
```

الفصل الثالث والأربعون: H_2 وH_2O وCO_2 — من المثال الجزيئي إلى ميزانية الكيوبتات

Chapter 43: H_2, H_2O, and CO_2 — From a Molecular Example to a Qubit Budget

43.1 لماذا H_2 هو نقطة البداية؟

43.1 Why H_2 Is the Starting Point

جزيء H_2 صغير بما يكفي لشرح دورة الكيمياء الكمومية كاملة: هندسة الجزيء، اختيار basis set، حساب مرجع Hartree–Fock، ثم التحويل إلى كيوبتات، بناء النموذج الإلكتروني، VQE والمقارنة مع مرجع كلاسيكي مثل FCI. هذا المثال لا يمثل كل الكيمياء الجزيئية.

H_2 is small enough to expose the full quantum-chemistry loop: molecular geometry, basis-set choice, a Hartree–Fock reference, an electronic model, qubit mapping, VQE, and comparison with a classical reference such as FCI. It does not represent all molecular chemistry.

القيمة العلمية للمثال تأتي من إمكانية تتبع كل قرار. إذا تغير طول الرابطة أو basis set أو active space، تتغير المسألة نفسها. لذلك لا يجوز مقارنة رقمين من تشغيلين مختلفين دون تسجيل هذه الإعدادات.

The scientific value comes from traceability of each decision. Changing bond length, basis set, or active space changes the problem itself. Two results from different configurations should therefore not be compared without recording those settings.

43.2 تنفيذ H_2O بفضاء نشط

43.2 Executing H_2O in an Active Space

الماء يحتوي إلكترونات ومدارات أكثر من H_2. لجعل المثال قابلًا للتشغيل على جهاز تعليمي صغير نختار active space محددًا ونذكر بوضوح أنه اختزال للمسألة. النتيجة ليست طاقة كل الإلكترونات في كل المدارات، بل نتيجة للنموذج المختار.

H_2O contains more electrons and orbitals than H_2. To make it executable on a small educational system, define an explicit active space and state that it is a reduction. The result is not the all-electron energy of every orbital; it is the result of the selected model.

يجب أن يسجل التقرير هندسة الجزيء، basis set، عدد الإلكترونات الفعالة، عدد المدارات الفعالة، عدد الكيوبتات، وخوارزمية المحسن. هذه الحقول تسمح للقارئ بإعادة بناء المسألة بدل التعامل مع رقم طاقة منفصل.

The report should store molecular geometry, basis set, active-electron count, active-orbital count, qubit count, and optimizer. These fields let the reader reconstruct the problem instead of treating the energy as an isolated number.

43.3 تنفيذ CO_2 بفضاء نشط

43.3 Executing CO_2 in an Active Space

ثاني أكسيد الكربون يقدم مثالًا أكثر تحديًا لأن عدد درجات الحرية يرتفع. لذلك نستخدم نفس المبدأ: تحديد model scope، ثم تنفيذ نموذج صغير، ثم مقارنة المرجع الكلاسيكي بالمحاكاة الكمومية. لا نخلط بين قابلية تشغيل النموذج وبين تمثيل كل درجات الحرية الفيزيائية.

CO_2 provides a more demanding example because the number of degrees of freedom grows. The same rule applies: define the model scope, execute a small model, and compare a classical reference

with the quantum simulation. Executability is not the same as representing every physical degree of freedom.

لكن عدد active space. لشرح لماذا يتزايد عدد الكيوبتات مع اتساع CO_2و H_2O يمكن استخدام الكيوبتات وحده ليس مقياسًا كافيًا؛ عمق الدائرة، عدد البوابات، الاتصال، الضوضاء، وعدد اللقطات كلها تدخل في تكلفة التنفيذ.

H_2O and CO_2 can show why qubit counts grow with active-space size. Qubit count alone is not enough: circuit depth, gate count, connectivity, noise, and shot count also determine execution cost.

43.4 ميزانية 4–6 كيوبت

43.4 A 4–6 Qubit Budget

النطاق 4–6 كيوبت مناسب للتعليم لأنه يسمح بتمثيل حالات متعددة دون أن يتحول المثال إلى مشروع محاكاة ضخم. عند 6 كيوبت يصبح فضاء الحالة 2^6 = 64 حالة أساس، وهو عدد صغير يمكن فحصه وفهمه، لكنه يكفي لشرح كيف ينمو فضاء الحالة أسيًا.

A 4–6 qubit range is useful educationally because it represents nontrivial states without becoming a large simulation project. At six qubits the state space contains 2^6 = 64 basis states, small enough to inspect yet sufficient to demonstrate exponential state-space growth.

يجب أن نميز بين عدد الكيوبتات المنطقية وعدد الكيوبتات الفيزيائية المطلوبة على جهاز حقيقي بعد الترجمة والتنفيذ مع تصحيح أو تخفيف الأخطاء. الرقم الموجود في النموذج لا يساوي بالضرورة الرقم الذي يحتاجه الجهاز.

Distinguish logical qubits in the model from physical resources required on real hardware after compilation and error-management choices. The qubit count in the model is not automatically the hardware resource count.

43.5 مقارنة FCI وHF وVQE

43.5 Comparing FCI, Hartree–Fock, and VQE

يعطي حلًا FCIيعطي تقريبًا منظمًا، و Hartree–Fock .المرجع الكلاسيكي يجب أن يكون واضحًا ansatz خوارزمية هجينة تعتمد على VQE أدق لمسألة صغيرة ضمن نفس فضاء الأساس، بينما ومحسن وقياسات. لا يصح مقارنة هذه الأرقام دون تحديد النموذج نفسه.

The classical references must be explicit. Hartree–Fock is a structured approximation, FCI is a more complete solution for a small problem within the same basis, while VQE is a hybrid algorithm depending on an ansatz, optimizer, and measurements. These values should not be compared without matching the model.

في مثال صغير فهذا دليل على نجاح ذلك الإعداد، وليس برهانًا على FCI من VQE إذا اقتربت نتيجة والضوضاء والمحسن ودقة القياس قبل ansatz تفوق كمومي عام. وإذا ابتعدت النتيجة، يجب فحص تفسير الفرق.

If VQE approaches FCI on a small example, that demonstrates success of that configuration, not general quantum advantage. If it differs, inspect the ansatz, noise, optimizer, and measurement precision before interpreting the gap.

PYTHON / qubit_budget.py

```
from __future__ import annotations

def basis_states(qubits: int) -> int:
    if not 1 <= qubits <= 30:
        raise ValueError("qubits must be between 1 and 30 for this planning calculation")
    return 2 ** qubits

for q in (2, 4, 6):
    print(q, basis_states(q))
```

PYTHON / classical_h2_fci_reference.py

```
from pyscf import gto, scf, fci

# Small H2 reference. H2O/CO2 runs in the project use explicit active spaces.
mol = gto.M(atom="H 0 0 0; H 0 0 0.74", basis="sto-3g", unit="Angstrom", charge=0, spin=0)
mf = scf.RHF(mol).run()
solver = fci.FCI(mol, mf.mo_coeff)
energy, _ = solver.kernel()
print({"hf_energy": float(mf.e_tot), "fci_energy": float(energy)})
```

الفصل الرابع والأربعون: الذكاء الاصطناعي التوليدي ونظام التوصية متعدد الأهداف

Chapter 44: Generative AI and a Multi-Objective Recommendation System

44.1 من التنبؤ إلى اقتراح المرشحين

44.1 From Prediction to Candidate Proposal

النموذج التوليدي لا يثبت أن المركب المقترح صالح. وظيفته إنشاء مرشحين في فضاء واسع ثم تمريرهم إلى طبقة تحقق مستقلة. في المشروع يمكن تمثيل المرشح بخصائص رقمية مع قيود قابلة للفحص قبل أي حساب مكلف.

A generative model does not prove that a proposed compound is valid. Its role is to generate candidates in a broad space and pass them to an independent validation layer. In this project, a candidate can be represented by numerical properties and explicit constraints before expensive computation.

عند استخدام VAE أو GAN يجب تسجيل مجموعة التدريب، إصدارها، عدد العينات المولدة، seed، ومعايير قبول المرشح. وعند استخدام LLM يجب أن يكون دور النموذج في صياغة الاستعلام أو تلخيص الأدلة، بينما تبقى قاعدة البيانات والمراجع طبقة الحقيقة.

When using a VAE or GAN, record the training dataset, its version, generated sample count, seed, and acceptance criteria. When using an LLM, keep its role to query formulation or evidence summarization while the database and references remain the truth layer.

44.2 نظام توصية متعدد الأهداف

44.2 A Multi-Objective Recommendation System

المرشح لا يملك أفضلية مطلقة. قد يكون منخفض الكثافة ومرتفع الثبات لكنه مكلف أو غير متاح. لذلك نخزن الأهداف والأوزان والقيود مع كل تشغيل، ونحتفظ بترتيب المرشحين بدل رقم واحد يخفي المفاضلات.

A candidate has no absolute best status. It may have low density and high stability but be costly or unavailable. Store objectives, weights, and constraints with every run and retain the ranking rather than hiding trade-offs in a single score.

بعد تطبيع المقاييس، لكن score = w1*property1 + w2*property2 يمكن أن يكون التقرير يجب أن يوضح طريقة التطبيع واتجاه كل خاصية. تغيير الأوزان يغير القرار، ولذلك يجب أن يكون تغيير الأوزان جزءًا من إصدار التجربة.

A score can be defined as w1*property1 + w2*property2 after normalization, but the report must state the normalization method and direction of each property. Changing weights changes the decision, so weights are part of the experiment version.

44.3 التحقق المستقل من المرشح

44.3 Independent Candidate Validation

بعد التوليد نعيد حساب الخواص بطريقة مستقلة أو نستخدم قاعدة مرجعية. إذا كان المرشح صالحًا عدديًا لكنه خارج نطاق التدريب، نسجل distribution shift بدل اعتباره نجاحًا. وإذا خالف قيدًا صريحًا يرفض قبل أي ادعاء علمي.

After generation, recompute properties independently or compare against a reference source. If a candidate is numerically valid but far outside the training distribution, record distribution shift rather than calling it a success. If it violates an explicit constraint, reject it before making a scientific claim.

التحقق المستقل هو الفاصل بين مولد مرشحين ونظام علمي. يمكن للنموذج أن يكرر أخطاء بيانات التدريب أو أن يستغل علاقة إحصائية لا تصمد عند القياس الحقيقي. لذلك لا يدخل المرشح مرحلة التوصية النهائية دون سجل تحقق.

Independent validation separates a candidate generator from a scientific system. A model can reproduce training-data errors or exploit statistical relationships that fail under measurement. A candidate therefore needs a validation record before final recommendation.

44.4 LLM وSQL بأمان تشغيلي

44.4 LLM and SQL with Operational Safety

عندما يطلب الباحث من نموذج لغوي استخراج عدد العناصر أو متوسط خاصية، يجب أن يحول السؤال إلى استعلام مضبوط ويُعرض الاستعلام قبل الاعتماد على النتيجة في المسار العلمي. لا نسمح للنموذج بكتابة SQL غير مقيدة ثم تنفيذها على قاعدة الإنتاج.

When a researcher asks a language model for a count or property average, the system should translate the question into a constrained query and expose the query before using the result scientifically. Do not allow an unconstrained generated SQL statement to run against a production database.

يمكن أن تمر الاستعلامات عبر قائمة جداول وأعمدة مسموحة، ثم فحص syntax، ثم تنفيذ read-only، ثم تسجيل query_hash والنتيجة. هذا يجعل النموذج اللغوي مساعدًا في الوصول إلى البيانات بدل أن يصبح طبقة غير قابلة للمراجعة فوق قاعدة البيانات.

Queries can pass through an allow-list of tables and columns, syntax validation, read-only execution, and logging of query_hash and result. The language model then assists data access without becoming an opaque layer over the database.

44.5 قياس الفائدة بدل عدّ الأدوات

44.5 Measure Capability, Not Tool Count

إضافة GAN أو VAE أو LLM ليست إنجازًا في حد ذاتها. السؤال التشغيلي هو: هل أصبح النظام قادرًا على استكشاف مرشحين أكثر، أو تسجيل سبب القبول والرفض، أو تقليل زمن الاستعلام، أو كشف أخطاء لم تكن ظاهرة؟ إذا لم تضف الأداة قدرة قابلة للقياس، فهي عبء.

Adding a GAN, VAE, or LLM is not an achievement by itself. The operational question is whether the system can explore more candidates, record acceptance and rejection reasons, reduce query time, or detect previously hidden errors. If the tool adds no measurable capability, it is a burden.

هذا المبدأ يحمي المشروع من تحويله إلى قائمة تقنيات. كل أداة يجب أن تدخل بسبب واضح، وتخرج بنتيجة قابلة للقياس، وتخضع لنفس قواعد provenance والاختبار والحوكمة التي تطبق على SQL والكم.

This principle prevents the project from becoming a technology list. Each tool should enter for a clear reason, produce a measurable capability, and follow the same provenance, testing, and governance rules as SQL and quantum workflows.

PYTHON / recommendation_gate.py

```python
from __future__ import annotations
from dataclasses import dataclass

@dataclass(frozen=True)
class Candidate:
    candidate_id: str
    property_a: float
    property_b: float
    valid: bool

def score(c: Candidate, w_a: float = 0.6, w_b: float = 0.4) -> float:
    if not c.valid:
        return float("-inf")
    return w_a * c.property_a + w_b * c.property_b

candidates = [Candidate("A", 0.80, 0.50, True), Candidate("B", 0.70, 0.90, True), Candidate("C", 0.95,
0.10, False)]
ranked = sorted(candidates, key=score, reverse=True)
print([(c.candidate_id, score(c)) for c in ranked])
```

الفصل الخامس والأربعون: اختبار الحجم الجينومي الاصطناعي — Big Data دون إدخال جينوم بشري

Chapter 45: Synthetic Genome-Scale Testing — Big Data Without Inserting a Human Genome

45.1 لماذا نستخدم بيانات اصطناعية؟

45.1 Why Use Synthetic Data?

الجينوم البشري الكامل ليس مادة مناسبة للحشو داخل هذا الكتاب. المطلوب هو أن يفهم القارئ كيف تتغير الهندسة عندما يصبح عدد الصفوف والملفات كبيرًا. لذلك ننشئ بيانات اصطناعية لها نفس خصائص الحجم والتجزئة والعلاقات دون إدخال معلومات شخصية.

A complete human genome is not appropriate as filler for this book. The goal is to understand how engineering changes when row counts, files, and relationships become large. Synthetic data therefore reproduce scale and structure without introducing personal information.

يمكن أن يمثل السجل الاصطناعي gene_id وvariant_id وsample_id وfeature_value وdataset_version. المهم أن تكون العلاقات بين الجداول واضحة وأن تحمل كل دفعة partition_id وsource_type=synthetic.

A synthetic record can contain gene_id, variant_id, sample_id, feature_value, and dataset_version. The important point is that table relationships remain explicit and every batch carries partition_id and source_type=synthetic.

45.2 إنشاء آلاف الملفات بدل ملف واحد ضخم

45.2 Many Partitions Instead of One Giant File

البيانات الكبيرة لا تعني فقط ملفًا أكبر. غالبًا توجد ملفات أو partitions يمكن قراءتها بالتوازي. هنا تظهر قيمة Parquet وDask وSpark، بينما يبقى SQLite مناسبًا للبيانات الصغيرة أو الميتاداتا التي تحتاج معاملات قوية.

Big data is not merely one larger file. Real workflows often contain many partitions that can be processed in parallel. This is where Parquet, Dask, and Spark become useful, while SQLite remains appropriate for small transactional metadata.

الاختيار بين Dask وSpark يعتمد على حجم البيانات، شكل العمليات، البنية التحتية، ومهارات الفريق. لا نستخدم Spark للجدول الدوري لأن 118 سجلًا لا تحتاج إلى نظام موزع؛ نستخدمه في مثال الحجم لإظهار متى يتغير القرار الهندسي.

Choosing Dask or Spark depends on data volume, workload shape, infrastructure, and team skills. Spark is not used for the 118-row periodic table because it does not need distributed infrastructure; it appears in the scale example to show when the engineering decision changes.

45.3 التحقق قبل التوزيع

45.3 Validate Before Distribution

توزيع بيانات خاطئة يضاعف سرعة الخطأ. لذلك يجب فحص الأنواع والمفاتيح والقيم المفقودة قبل التقسيم، ثم فحص كل partition بعد كتابته. تحفظ أعداد الصفوف والبصمات ونسخة المخطط لكل دفعة.

Distributing bad data only makes the error faster. Validate types, keys, and missingness before partitioning, then validate each partition after writing it. Store row counts, hashes, and schema versions for every batch.

يمكن تشغيل اختبارات صغيرة على عينات ثم اختبارات شاملة قبل الإصدار. إذا اختلف عدد الصفوف المتوقع عن المجموع الفعلي، يتوقف خط الإنتاج. هذه قاعدة بسيطة لكنها ضرورية عندما يصبح التصحيح بعد الكتابة مكلفًا.

Small-sample tests can run early, followed by full checks before release. If expected row counts do not match the total, the pipeline stops. This simple rule becomes essential when post-write correction is expensive.

45.4 من الجين إلى البروتين إلى الدليل

45.4 From Gene to Protein to Evidence

عند ربط العناصر بالبروتينات والإنزيمات نستخدم مفاتيح واضحة: gene_id، protein_id، element_symbol، evidence_id. وجود عنصر في بروتين لا يثبت أنه سبب مرض أو مقاومة؛ هو رابط كيميائي أو بنيوي يحتاج إلى سياق بيولوجي مستقل.

When linking elements to proteins and enzymes, use explicit keys such as gene_id, protein_id, element_symbol, and evidence_id. The presence of an element in a protein does not prove that it causes disease or resistance; it is a chemical or structural link that requires independent biological context.

في هذا المشروع Gene Resistance Tracker طبقة تنظيم أدلة، لا نظامًا لتعديل الجينات أو لتقديم قرار علاجي. هذا الفصل بين تنظيم البيانات وبين الاستنتاج السريري يجب أن يبقى واضحًا في كل واجهة وكود.

The Gene Resistance Tracker is an evidence-organization layer, not a gene-editing system or a clinical decision system. This separation between data organization and clinical inference must remain explicit in every interface and code path.

45.5 اختبار الضغط وتفسير النتائج

45.5 Stress Testing and Interpreting Results

اختبار الحجم لا يسأل فقط كم استغرق التنفيذ. نسجل throughput، peak memory، عدد partitions، حجم كل ملف، زمن القراءة، زمن التحويل، وزمن الكتابة. بهذه القياسات يمكن معرفة أين توجد عنق الزجاجة بدل تخمينها.

Scale testing is not only about elapsed time. Record throughput, peak memory, partition count, file size, read time, transformation time, and write time. These measurements reveal bottlenecks instead of relying on guesses.

إذا تحسن الزمن لأننا استخدمنا جهازًا أقوى فقط، فهذا ليس بالضرورة تحسنًا في الخوارزمية. يجب مقارنة الإعدادات مع تثبيت حجم البيانات والعمليات والنسخة البرمجية قدر الإمكان، ثم توثيق أي اختلاف في العتاد.

If runtime improves only because a stronger machine was used, that is not necessarily an algorithmic improvement. Keep data size, operations, and software version controlled when possible and record hardware differences.

PYTHON / synthetic_genome_partition_validator.py

```
from __future__ import annotations
import pandas as pd
from pathlib import Path

REQUIRED = {"gene_id", "variant_id", "sample_id", "feature_value", "dataset_version", "source_type"}

def validate_partition(path: str) -> dict:
    df = pd.read_parquet(Path(path))
    missing = REQUIRED - set(df.columns)
    if missing:
        raise ValueError(f"Missing columns: {sorted(missing)}")
    if set(df["source_type"].dropna()) - {"synthetic"}:
        raise ValueError("This benchmark accepts synthetic source_type only")
    return {"rows": int(len(df)), "columns": list(df.columns)}

if __name__ == "__main__":
    print("Synthetic partition validator ready.")
```

الخاتمة والتوصيات

Conclusion & Recommendations

الخاتمة

Conclusion

في نهاية هذا الجزء تكون قد اتبعت مسارًا واضحًا: بدأت بتمثيل الجدول الدوري كبيانات منظمة، ثم بنيت مخطط SQL، واستخرجت البيانات إلى مصفوفات NumPy، وأنشأت مستودع ميزات للتعلم الآلي، ثم انتقلت إلى تمثيلات كمومية وحالة VQE لجزيء الهيدروجين H_2. كل خطوة هنا مثال قابل للفحص، وليست ادعاءً بأن هذه الأدوات وحدها تمثل كل مسائل الكيمياء أو البيولوجيا.

By the end of this part, you have followed a clear path: structured periodic-table data, an SQL schema, NumPy matrices, a feature store for machine learning, quantum-state representations, and a VQE workflow for molecular hydrogen H_2. Each step is a testable example, not a claim that these tools alone represent all problems in chemistry or biology.

ما أنجزته:

What you have built:

1. قاعدة بيانات متكاملة للجدول الدوري تحتوي على 118 عنصرًا بخصائصها العلمية الدقيقة.

1. A normalized periodic-table database containing the documented identities of 118 elements.

2. مصفوفات رقمية جاهزة للتحليل العددي والتعلم الآلي.

2. Numerical matrices ready for numerical analysis and machine learning.

3. مستودع ميزات (Feature Store) يمكن استخدامه في تدريب نماذج الذكاء الاصطناعي.

3. A Feature Store that supports documented machine-learning workflows.

4. حالات كمومية تمثل خصائص العناصر، مخزنة في قاعدة البيانات وجاهزة للمعالجة الكمومية.

4. Quantum states representing encoded element features, stored in the database for quantum-processing workflows.

5. محاكاة VQE لجزيء الهيدروجين H_2، تربط البيانات الكلاسيكية بالحوسبة الكمومية.

5. A VQE simulation of molecular hydrogen H_2 that connects classical data with quantum computing.

التوصيات للمستقبل (الجزء الرابع)

Future Directions (Part Four)

بناءً على ما أنجزناه، يمكن تطوير المشروع في الاتجاهات التالية:

Based on the current system, the project can be developed along the following controlled directions:

1. تطوير المحاكاة الكمومية

1. Quantum simulation expansion

· محاكاة جزيئات أكثر تعقيدًا (مثل H_2، H_2O، CO_2)

• Simulate larger molecules such as H_2O or other systems only after constructing their physical Hamiltonians.

· استخدام دوائر كمومية أكبر (2–4 كيوبت) لتمثيل أنظمة متعددة الإلكترونات.

• Use larger quantum circuits (2–4 qubits) to represent multi-electron systems.

· دراسة HHL بوصفها خوارزمية لأنظمة المعادلات الخطية، مع توضيح كيف يمكن أن تدخل في صياغات فيزيائية مناسبة دون تقديمها كحل مباشر لمعادلة شرودنجر.

• Explore advanced quantum algorithms such as HHL only when their assumptions and numerical requirements are explicitly satisfied.

2. الذكاء الاصطناعي التوليدي

2. Generative Artificial Intelligence

· استخدام GANs أو VAEs لاستكشاف فضاء مرشحين لمواد أو تراكيب مع تقييم مستقل للاستقرار والخواص، دون اعتبار المخرجات حقائق تجريبية.

• Use GANs or VAEs to explore candidate materials only as hypothesis-generation tools, not as experimental evidence.

· تدريب نماذج لغوية كبيرة (LLMs) لاقتراح مركبات جديدة على بيانات الجدول الدوري.

• Train large language models (LLMs) on periodic-table data to propose candidate compounds, while keeping proposals distinct from experimental evidence.

· تطوير نظام توصية (Recommendation System) لاقتراح خليط من العناصر لتحقيق خواص محددة.

• Develop a recommendation system for candidate mixtures while treating its outputs as computational hypotheses requiring independent validation.

3. التكامل مع البيانات الجينومية

3. Genomic-data Integration

متتبع مقاومة الجينات

Gene Resistance Tracker

تظهر هذه اللوحة كنموذج مخطط بيانات مفاهيمي لمسار التكامل الجينومي المستقبلي: عينات وتسلسلات وطفرات وأدلة ونتائج، وليست بروتوكولًا تجريبيًا أو تعليمات تعديل جيني.

This plate is a conceptual data-schema example for future genomic integration: samples, sequences, mutations, evidence, and results. It is not an experimental or gene-editing protocol.

· ربط بيانات الجدول الدوري بالبيانات الجينومية (مثل وجود العناصر في البروتينات والإنزيمات)

• Link element data to genomic or proteomic datasets only through well-defined schemas and lawful, privacy-preserving workflows.

· تحليل العلاقة بين وفرة العناصر في الجسم البشري والأمراض الجينية

• Treat any association between elemental abundance and disease as a hypothesis requiring independent biomedical evidence.

4. التوسع في الحوكمة الأخلاقية

4. Ethical Governance Expansion

· إضافة طبقات إضافية من الحوكمة لضمان استخدام التصاميم التوليدية بمسؤولية.

• Add governance gates for generated designs and computational proposals.

· تطوير نظام "موافقة مسبقة" (Pre-Approval) لأي محاكاة كمومية أو اقتراح توليدي.

• Require documented review before high-impact simulations or automated recommendations are acted upon.

· ربط النتائج الحاسوبية بسجلات قرارات قابلة للتدقيق

• Link computational results to auditable decision records

· ربط النتائج الكمومية بسجلات التدقيق الأخلاقي (كما في الجزء الثاني والثالث)

• Link quantum-computing results to auditable ethical-review records when appropriate.

5. الأدوات والتقنيات

5. Tools and Technologies

· استخدام Dask أو Spark للتعامل مع كميات أكبر من البيانات

• Use Dask or Spark when the dataset size genuinely requires distributed processing.

· دمج PyTorch أو TensorFlow للنماذج العميقة

• Integrate PyTorch or TensorFlow when deep-learning models are actually required by the task.

· استخدام Qiskit Runtime أو AWS Braket للتشغيل على أجهزة كمومية حقيقية.

• Use Qiskit Runtime or AWS Braket only when a real hardware experiment is intended and its cost, noise, and provenance are documented.

· أدوات لاستخراج بيانات الجدول الدوري وتحليلها

• Tools for extracting and analyzing periodic-table data

أو إلى Qiskit Runtime، المحلي إلى خدمات مثل StatevectorEstimator يمكن أن تنتقل التجارب من
عند هذه النقطة تظهر عناصر جديدة يجب تسجيلها: الجهاز، النسخة البرمجية، AWS Braket. أجهزة متاحة عبر

عدد القياسات، خصائص الضوضاء، وأي أسلوب لتخفيف الخطأ. لذلك لا ينبغي أن ،transpilation الدائرة بعد
.تعرض نتيجة عتاد حقيقي باعتبارها امتدادًا تلقائيًا لنتيجة المحاكاة المثالية؛ يجب أن تحمل معها سجل تنفيذها ومصدرها

Experiments can move from local StatevectorEstimator simulations to services such as Qiskit Runtime or hardware available through AWS Braket. At that point, new provenance fields must be recorded: device, software version, transpiled circuit, shot count, noise characteristics, and any error-mitigation method. A hardware result should therefore not be presented as an automatic continuation of an ideal simulation result; it should carry its own execution and provenance record.

الوصول إلى العتاد الكمومي الحقيقي

Moving from simulation to real quantum hardware.

لتقسيم العمل. Spark أو Dask عندما يتجاوز حجم البيانات الذاكرة أو تصبح العمليات المتكررة بطيئة، يمكن استخدام
ولا ينبغي استخدام التوزيع لمجرد أنه يبدو أكثر تقدمًا؛ إذا كانت البيانات صغيرة فحل أبسط قد يكون أفضل. وبالمثل،
عندما تتطلب المهمة نموذجًا عميقًا فعليًا، مع حفظ تقسيم التدريب TensorFlow أو PyTorch يمكن إدخال
.والاختبار، وإصدارات البيانات، ومقاييس الأداء حتى يمكن إعادة إنتاج النتيجة

When data no longer fits comfortably in memory or repeated computation becomes slow, Dask or Spark can partition the workload. Distributed processing should not be used merely because it looks more advanced; for small data, a simpler solution may be better. Likewise, PyTorch or TensorFlow should be introduced when the task genuinely requires deep learning, while preserving train/test splits, data versions, and performance metrics so the result remains reproducible.

.والتعلم العميق Sparkو Dask :التوسع الحسابي

Computational scaling: Dask, Spark, and deep learning.

عندما تصبح الأنظمة التوليدية أو الكمومية قادرة على إنتاج عدد كبير من الاقتراحات، تصبح المراجعة جزءًا من خط
الأنابيب نفسه. يمكن أن تتضمن البنية بوابة موافقة مسبقة، وسجلًا للمدخلات والإصدارات، وسببًا لرفض أو قبول
النتيجة، وربطًا بين كل نتيجة وحسابها ومصدر بياناتها. هذه الطبقات ليست عبارات فلسفية خارج النظام؛ إنها آليات
.هندسية تقلل احتمال استخدام نتيجة غير موثقة في قرار ذي أثر كبير

When generative or quantum workflows can produce many candidate results, review becomes part of the pipeline itself. The architecture can include a pre-approval gate, records of inputs and versions, reasons for accepting or rejecting results, and links between every result, its computation, and its data provenance. These layers are not philosophical statements outside the system; they are engineering mechanisms that reduce the chance that an unverified result is used in a high-impact decision.

الحوكمة قبل التوليد والتنفيذ

Governance before generation and execution

يمكنك ربط سجلات العناصر ببيانات البروتينات والإنزيمات، ثم إضافة طبقات مثل العينات والتسلسلات والمتغيرات والأدلة ونتائج التحليل. لكن هذا لا يعني أن الجدول الدوري يصبح جينومًا، ولا أن وجود عنصر في بروتين يثبت علاقة سببية بمرض. في هذا الجزء يبقى Gene Resistance Tracker نموذجًا مفاهيميًا لهندسة البيانات والتتبع، ويمكن تطويره لاحقًا ببيانات حيوية مصرح بها ومراجعة مستقلة.

You can link element records to protein and enzyme data and then add layers such as samples, sequences, variants, evidence, and analysis results. This does not turn the periodic table into a genome, and the presence of an element in a protein does not establish a causal relationship with disease. Here, the Gene Resistance Tracker remains a conceptual model for data engineering and provenance, to be extended later with authorized biological datasets and independent review.

التكامل الجينومي بوصفه جسرًا لا اختصارًا.

Genomic integration as a bridge, not a shortcut.

أو نماذج توليد أخرى. لكن VAEs أو GANs يمكن أن تتحول بيانات العناصر والمواد إلى مساحة اقتراحات باستخدام النموذج التوليدي لا يثبت أن المادة المقترحة قابلة للتصنيع أو مستقرة أو آمنة؛ إنه يولد فرضيات تحتاج إلى حسابات مستقلة وبيانات مرجعية وتجارب تحقق. وينطبق الأمر نفسه على LLMs: يمكنها اقتراح مركبات أو تفسير سجلات، لكنها لا تستبدل قواعد الكيمياء أو التحقق التجريبي.

Element and materials data can be turned into a candidate-generation space using GANs, VAEs, or other generative models. A generative model does not prove that a proposed material is synthesizable, stable, or safe; it generates hypotheses that require independent calculations, reference data, and validation. The same distinction applies to LLMs: they can propose compounds or interpret records, but they do not replace chemical constraints or experimental verification.

تطوير الذكاء الاصطناعي التوليدي

Expanding generative artificial intelligence

وعند ذكر HHL، ينبغي أن تتعامل معه بوصفه خوارزمية لحل أنظمة خطية كمومية، لا بوصفه زرًا مباشرًا لحل معادلة شرودنجر. لكي يدخل في مسار فيزيائي، يجب أولًا صياغة المسألة في صورة نظام خطي مناسب، مع مراعاة شروط مثل بنية المصفوفة، والتهيئة، وحالة الشرط، وطريقة استخراج المعلومة من الحالة الكمومية. لذلك يبقى HHL في هذا المشروع موضوعًا متقدمًا للتوسّع، وليس نتيجة جاهزة من المثال الحالي.

When HHL is discussed, treat it as a quantum linear-system algorithm rather than as a direct button for solving the Schrödinger equation. A physical application first requires a suitable linear-system formulation, together with attention to matrix structure, state preparation, conditioning, and how information is extracted from the quantum state. In this project, HHL is therefore an advanced extension topic rather than a completed result of the present example.

فلا يكفي زيادة عدد الكيوبتات شكليًا. يجب أولًا بناء نموذج إلكتروني محدد، CO_2 أو H_2O إلى H_2 إذا انتقلت من واختيار قاعدة أساس وتقريب فيزيائي مناسب، ثم تحويل الهاملتوني إلى تمثيل كيوبتي يمكن تنفيذه. توجد دراسات فعلية لمحاكاة H_2O وCO_2 باستخدام VQE، لكنها توضح أيضًا أن الموارد المطلوبة تنمو بسرعة وأن اختيار الاختزال والـactive space يؤثر في حجم الدائرة والنتيجة.

Moving from H_2 to H_2O or CO_2 is not simply a matter of increasing the qubit count. A defined electronic-structure model, basis set and physical approximation must be chosen first, followed by a mapping of the Hamiltonian to a qubit representation. Published VQE studies

have simulated H_2O and CO_2, but they also show that resource requirements grow quickly and that choices such as active-space reduction affect circuit size and accuracy.

ملاحظة علمية حول المحاكاة الكمومية

Scientific note on quantum simulation

كلمة أخيرة

Final Note

هذا المشروع يجمع أمثلة مترابطة في هندسة البيانات العلمية: قاعدة بيانات، وتحليل عددي، وتعلم آلي، وتمثيل كمومي، وتخزين للنتائج مع تتبع للمصدر والإصدار. المقصود أن ترى كيف تتكامل هذه الطبقات عندما تصبح البيانات أكبر وأكثر تعقيدًا، وكيف تبقى القيود العلمية والأخلاقية جزءًا من التصميم.

This project brings together connected examples in scientific data engineering: a database, numerical analysis, machine learning, quantum representation, and persistent result storage with provenance and versioning. The purpose is to show how these layers fit together as data and computational complexity grow, while scientific and ethical constraints remain part of the design.

أنت الآن تمتلك:

You now have:

• قاعدة بيانات كاملة للجدول الدوري

• A complete periodic-table database

• أدوات لاستخراجها وتحليلها

• Tools for extracting and analyzing the data

• بنية تحتية للذكاء الاصطناعي

• An AI infrastructure with explicit feature provenance

• جسرًا نحو الحوسبة الكمومية

• A bridge toward quantum computing

المستقبل بين يديك. استخدم هذه الأدوات لبناء تطبيقات تحل مشاكل حقيقية في الكيمياء، والبيولوجيا، وعلوم المواد، والطب.

The next step is yours: use these tools to build reproducible applications for real questions in chemistry, biology, materials science, and medicine.

يمكنك الآن الانتقال إلى الجزء التالي من الرحلة التعليمية باستخدام هذه الأدوات في مسائل أكثر واقعية، مع الحفاظ على قابلية إعادة الإنتاج، وحدود الادعاء العلمي، وسلامة البيانات والقرارات.

You can now carry these tools into more realistic problems while preserving reproducibility, clear scientific claims, data integrity, and responsible decision-making.

PYTHON

```
checks = {
    "tests": True,
    "118_elements": True,
    "searchable_pdf": True,
    "no_replacement_chars": True,
    "images_intact": True,
    "toc_updated_last": True,
}
assert all(checks.values()), checks
print("RELEASE GATE: PASS")
```

38.10 بوابة الإصدار النهائي

38.10 Final Release Gate

PYTHON

```
def require_column(columns, name):
    if name not in columns:
        raise ValueError(f"Missing required column: {name}")

try:
    require_column([], "atomic_number")
except ValueError as exc:
    print("EXPECTED FAILURE:", exc)
```

اختبار النظام لا يكتمل باختبار المسار الناجح فقط. يجب إدخال فشل متعمد: ملف مفقود، عمود غير موجود، إصدار غير متوافق، قيمة خارج النطاق، أو خدمة كمومية غير متاحة. يجب أن يفشل النظام بطريقة مفهومة وأن يسجل المرحلة والسبب، وألا يكتب نتيجة مزيفة. القدرة على قول «لم تُحسب النتيجة» أكثر أمانًا من إنتاج رقم يبدو معقولًا لكنه غير موثوق.

System testing is not complete if only the successful path is tested. Introduce deliberate failures: a missing file, absent column, incompatible version, out-of-range value, or unavailable quantum

service. The system should fail clearly, record the stage and reason, and avoid writing a fabricated result. The ability to state “the result was not computed” is safer than producing a plausible-looking number with no trustworthy provenance.

38.9 اختبار فشل متعمد

38.9 Deliberate Failure Testing

PYTHON

```python
from pathlib import Path

required = [
    "H₂O", "CO₂", "HHL",
    "EstimatorV2", "Gene Resistance Tracker",
    "requirements.txt", "Table of Contents",
]
text = Path(
    "Brick_by_Brick_Part_III_MASTER.txt"
).read_text(encoding="utf-8")
missing = [x for x in required if x not in text]
raise SystemExit(missing or "CONTENT GATE: PASS")
```

BASH

```bash
set -euo pipefail

python run_pipeline.py --stage database
python -m pytest -q
python run_pipeline.py --stage ml
python run_pipeline.py --stage quantum --backend local
python run_pipeline.py --stage report

# Cloud execution is explicit and never automatic.
# python run_pipeline.py --stage quantum --backend ibm --approved true
```

يفضل أن يكون التشغيل قابلًا للتجزئة حتى يختبر القارئ طبقة قبل الانتقال إلى التالية. تبدأ database ببناء المخطط وإدخال العناصر، ثم ml تبني الميزات وتنفذ التحقق، ثم quantum تشغل الأمثلة محليًا أو ترسلها إلى خدمة حقيقية إذا كانت البيئة مهيأة، ثم report تنتج المخرجات. كل مرحلة تعيد رمز خروج مناسبًا وتسجل الإصدار والمدخلات والمخرجات.

Execution should be staged so the reader can verify one layer before moving to the next. The database stage builds the schema and loads elements; the ML stage builds features and validation outputs; the quantum stage runs local examples or submits to a real service when configured; and the report stage produces outputs. Each stage

returns an exit code and records software version, inputs, and outputs.

ملحق تشغيلي: تشغيل المراحل والتحقق

Operational Appendix: Running and Verifying the Stages

TEXT

```
brick-by-brick-part-iii/
├── data/
│   ├── elements.csv
│   └── synthetic_genomics/
├── scripts/
│   ├── schema.sql
│   ├── load_elements.py
│   └── run_pipeline.py
├── src/
│   ├── models.py
│   ├── features.py
│   └── quantum/
├── tests/
│   ├── test_database.py
│   ├── test_features.py
│   └── test_release_gate.py
├── reports/
├── Dockerfile
├── requirements.txt
├── environment.yml
└── README.md
```

النسخة القابلة للنشر يجب أن تجعل ملفات النظام مفهومة: قاعدة البيانات منفصلة عن السكربتات والاختبارات والتقارير والبيانات الاصطناعية. run_pipeline.py ينسق المراحل، وrequirements.txt يثبت البيئة، ومجلد tests يثبت العقود. لا تحفظ مفاتيح الخدمات داخل المستودع، ولا تضع بيانات صحية شخصية في المثال.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts. Cloud credentials must not be committed, and personal health data do not belong in the educational example.

ملحق تشغيلي: بنية المشروع

Operational Appendix: Project Structure

ملحق المصدر التشغيلي: الملفات كما ينبغي أن تحفظ

Operational Source Appendix: Files as They Should Be Kept

هذا الملحق يربط الفصول بالملفات التي يمكن أن تمثل المشروع فعليًا. الغرض ليس تحويل الكتاب إلى مستودع كامل، بل إعطاء القارئ نقاط دخول واضحة: مخطط SQL، نماذج OOP، خط الميزات، التقييم، الأمثلة الكمومية، البيانات الجينومية الاصطناعية، الحوكمة، الاختبارات، الواجهة، والبيئة. كل ملف يحمل مسؤولية محددة، ويمكن اختباره أو استبداله دون إعادة كتابة بقية النظام.

This appendix connects the chapters to source files that can represent the project operationally. It is not intended to turn the book into a complete repository; instead, it gives the reader clear entry points: SQL schema, OOP models, feature construction, evaluation, quantum examples, synthetic genomic data, governance, tests, interface, and environment. Each file has a defined responsibility and can be tested or replaced without rewriting the rest of the system.

TEXT

```
The source tree is organized by responsibility:

data/               input and synthetic datasets
scripts/            orchestration and database loading
src/                domain and analytical code
tests/              software and data-contract tests
reports/            generated outputs
Dockerfile          reproducible container build
requirements.txt    pinned Python environment
environment.yml     Conda environment
README.md           operational documentation

The repository must not contain cloud credentials or
personal health data. Real-hardware access is a separate
approved execution path.
```

ملف المصدر: ci/scientific-ci.yml — التكامل المستمر والتحقق العلمي

Source File: ci/scientific-ci.yml — CI prevents software contracts from breaking. Real-hardware tests remain explicit and are not automatically run on every commit.

CI يمنع كسر العقود البرمجية. اختبارات العتاد الحقيقي تبقى مرحلة صريحة ولا تُشغَّل تلقائيًا في كل commit.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts. Cloud credentials must not be committed, and personal health data do not belong in the educational example.

YAML

```yaml
name: scientific-ci

on:
  push:
  pull_request:

jobs:
  test:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4

      - uses: actions/setup-python@v5
        with:
          python-version: "3.12"

      - name: Install
        run: |
          python -m pip install --upgrade pip
          pip install -r requirements.txt

      - name: Unit tests
        run: python -m pytest -q

      - name: Content contract
        run: |
          python scripts/check_release_content.py
```

ملف المصدر: requirements.txt — نسخة البيئة البرمجية المثبتة

Source File: requirements.txt — Pinned versions make environment differences explainable. Any upgrade should be deliberate, documented, and followed by full testing.

النسخ المثبتة تجعل اختلاف البيئة قابلًا للتفسير. أي تحديث يجب أن يكون قرارًا موثقًا يعقبه اختبار كامل.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts. Cloud credentials must not be committed, and personal health data do not belong in the educational example.

TXT / requirements.txt

```
numpy==2.3.5
pandas==2.2.3
scikit-learn==1.8.0
qiskit==2.5.0
qiskit-algorithms==0.4.0
qiskit-nature==0.8.0
pyscf==2.14.0
pytest==8.4.1
dask==2026.7.1
qiskit-ibm-runtime==0.47.0
```

ملف المصدر: Dockerfile — بيئة تشغيل قابلة لإعادة البناء

Source File: Dockerfile — Docker runs tests before the image is treated as valid and uses pinned requirements.

يشغل Docker الاختبارات قبل اعتماد الصورة، ويستخدم ملف المتطلبات المثبت.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts.

Cloud credentials must not be committed, and personal health data do not belong in the educational example.

DOCKERFILE

```
FROM python:3.12-slim

ENV PYTHONDONTWRITEBYTECODE=1
ENV PYTHONUNBUFFERED=1

WORKDIR /app

COPY requirements.txt .
RUN python -m pip install \
    --no-cache-dir -r requirements.txt

COPY . .

RUN python -m pytest -q

CMD [
    "python",
    "scripts/run_pipeline.py",
    "--stage",
    "all"
]
```

ملف المصدر: app.py — واجهة Streamlit قابلة للمراجعة

Source File: app.py — The interface displays results with run_id and data_version instead of isolated numbers.

الواجهة تعرض النتائج مع run_id وdata_version بدل تقديم رقم منفصل عن سياقه.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts. Cloud credentials must not be committed, and personal health data do not belong in the educational example.

PYTHON

```
import sqlite3
import streamlit as st

st.set_page_config(
    page_title="Brick by Brick Part III",
    layout="wide",
)

st.title("Brick by Brick — Part III")
```

```
st.caption(
    "Educational example; results require "
    "scientific interpretation."
)

con = sqlite3.connect("data/project.db")
count = con.execute(
    "SELECT COUNT(*) FROM elements"
).fetchone()[0]

st.metric("Elements", count)

run_id = st.text_input(
    "Run identifier", value="demo-001"
)
data_version = st.text_input(
    "Data version", value="elements-v1"
)

st.write({
    "run_id": run_id,
    "data_version": data_version,
    "status": "display-only",
})
```

ملف المصدر: tests/test_integrity.py — اختبارات تكامل البيانات الأساسية

Source File: tests/test_integrity.py — Tests prevent structural failures such as missing elements, duplicate keys, or invalid atomic weights.

الاختبارات تمنع أخطاء بنيوية مثل فقدان عنصر أو تكرار مفتاح أو وزن ذري غير صالح.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts. Cloud credentials must not be committed, and personal health data do not belong in the educational example.

PYTHON

```
import sqlite3

def test_element_count():
    con = sqlite3.connect("data/project.db")
    count = con.execute(
        "SELECT COUNT(*) FROM elements"
    ).fetchone()[0]
    assert count == 118

def test_atomic_numbers_unique():
    con = sqlite3.connect("data/project.db")
```

```
    rows = con.execute(
        "SELECT atomic_number FROM elements"
    ).fetchall()
    values = [r[0] for r in rows]
    assert len(values) == len(set(values))
    assert set(values) == set(range(1, 119))

def test_weights_positive():
    con = sqlite3.connect("data/project.db")
    bad = con.execute(
        "SELECT COUNT(*) FROM elements "
        "WHERE atomic_weight <= 0"
    ).fetchone()[0]
    assert bad == 0
```

ملف المصدر: scripts/run_pipeline.py — نقطة الدخول الموحدة للمشروع

Source File: scripts/run_pipeline.py — The single entry point prevents arbitrary stage order and allows one stage to be rerun and failures to be diagnosed.

نقطة الدخول تمنع تشغيل المراحل بترتيب عشوائي وتتيح إعادة تنفيذ مرحلة واحدة وتشخيص الفشل.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts. Cloud credentials must not be committed, and personal health data do not belong in the educational example.

PYTHON

```python
import argparse
import subprocess
import sys

def run(cmd):
    print("RUN:", " ".join(cmd))
    result = subprocess.run(cmd, check=False)
    if result.returncode:
        raise SystemExit(result.returncode)

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument(
        "--stage",
        choices=[
            "database", "ml", "quantum",
            "report", "all"
        ],
        required=True,
    )
    parser.add_argument("--backend", default="local")
```

```
    args = parser.parse_args()

    if args.stage in {"database", "all"}:
        run([sys.executable, "scripts/load_elements.py"])

    if args.stage in {"ml", "all"}:
        run([sys.executable, "scripts/run_ml.py"])

    if args.stage in {"quantum", "all"}:
        run([
            sys.executable,
            "scripts/run_quantum.py",
            "--backend", args.backend,
        ])

    if args.stage in {"report", "all"}:
        run([sys.executable, "scripts/build_report.py"])

if __name__ == "__main__":
    main()
```

ملف المصدر: src/governance.py — منطق الموافقة وسجل القرار

Source File: src/governance.py — This file turns governance into an operational state; a request does not enter execution before its conditions are complete.

هذا الملف يحول الحوكمة إلى حالة تشغيلية؛ الطلب لا يدخل التنفيذ قبل اكتمال شروطه.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts. Cloud credentials must not be committed, and personal health data do not belong in the educational example.

PYTHON

```
from dataclasses import dataclass
from datetime import datetime, timezone
import json

@dataclass
class Request:
    request_id: str
    purpose: str
    risk: str
    data_version: str
    approved_by: str | None = None
    status: str = "pending"

    def approve(self, actor):
        if not self.purpose:
```

```
            raise ValueError("purpose is required")
        if self.risk not in {"low", "medium", "high"}:
            raise ValueError("invalid risk")
        self.approved_by = actor
        self.status = "approved"

    def audit_payload(self):
        return json.dumps({
            "request_id": self.request_id,
            "purpose": self.purpose,
            "risk": self.risk,
            "data_version": self.data_version,
            "approved_by": self.approved_by,
            "status": self.status,
            "created_at": datetime.now(
                timezone.utc
            ).isoformat(),
        }, ensure_ascii=False)
```

ملف المصدر: src/dask_pipeline.py — خط Dask لمعالجة الأجزاء

Source File: src/dask_pipeline.py — The workflow uses map_partitions so the full dataset is not collected into memory before processing.

يستخدم المسار map_partitions حتى لا يجمع كل البيانات في الذاكرة قبل المعالجة.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts. Cloud credentials must not be committed, and personal health data do not belong in the educational example.

PYTHON

```
import dask.dataframe as dd

def add_variant_key(df):
    df = df.copy()
    df["variant_key"] = (
        df["chromosome"].astype(str)
        + ":"
        + df["position"].astype(str)
        + ":"
        + df["ref"]
        + ">"
        + df["alt"]
    )
    return df

def run():
    ddf = dd.read_parquet(
```

```
        "data/synthetic_genomics/*.parquet"
    )
    enriched = ddf.map_partitions(add_variant_key)
    return (
        enriched.groupby("chromosome")
        .size()
        .compute()
    )

if __name__ == "__main__":
    print(run())
```

ملف المصدر: src/genomic_synthetic.py — بيانات جينومية اصطناعية كبيرة

Source File: src/genomic_synthetic.py — This data does not represent a real person. Its purpose is to test scale, memory, joins, and partitioning.

هذه البيانات لا تمثل شخصًا حقيقيًا. وظيفتها اختبار الحجم والذاكرة والربط والتجزئة.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts. Cloud credentials must not be committed, and personal health data do not belong in the educational example.

PYTHON

```python
from pathlib import Path
import numpy as np
import pandas as pd

OUT = Path("data/synthetic_genomics")
OUT.mkdir(parents=True, exist_ok=True)

def make_chunk(seed, rows=250_000):
    rng = np.random.default_rng(seed)
    return pd.DataFrame({
        "sample_id": rng.integers(1, 100_001, rows),
        "chromosome": rng.integers(1, 23, rows),
        "position": rng.integers(1, 250_000_001, rows),
        "ref": rng.choice(list("ACGT"), rows),
        "alt": rng.choice(list("ACGT"), rows),
    })

def generate(chunks=8):
    for i in range(chunks):
        make_chunk(i).to_parquet(
            OUT / f"chunk-{i:03d}.parquet",
            index=False,
        )
```

```
if __name__ == "__main__":
    generate()
    print("synthetic genomic dataset generated")
```

ملف المصدر: src/hhl_demo.py — تطبيق HHL على نظام خطي صغير

Source File: src/hhl_demo.py — The file defines the classical reference first. A full HHL circuit must record phase-estimation precision, state preparation, controlled rotation, and readout assumptions.

الملف يعرّف أولًا المرجع الكلاسيكي. بناء دائرة HHL الفعلية يحتاج تسجيل دقة تقدير الطور والتهيئة والدوران المشروط والاستخراج.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts. Cloud credentials must not be committed, and personal health data do not belong in the educational example.

PYTHON

```python
import numpy as np

def classical_reference():
    A = np.array([
        [1.0, 0.0],
        [0.0, 3.0],
    ])
    b = np.array([1.0, 1.0])
    return A, b, np.linalg.solve(A, b)

def condition_number(A):
    return float(np.linalg.cond(A))

if __name__ == "__main__":
    A, b, x = classical_reference()
    print("A =", A)
    print("b =", b)
    print("x =", x)
    print("condition_number =", condition_number(A))
    print(
        "HHL requires explicit assumptions for "
        "state preparation, phase precision, and post-selection."
    )
```

ملف المصدر: src/fci_reference.py — مرجع FCI للمقارنة الكلاسيكية

Source File: src/fci_reference.py — FCI is a reference for a small problem, not a claim that it is always better. An independent reference protects comparisons from unsupported interpretation.

FCI مرجع لمسألة صغيرة وليس ادعاءً بأنه أفضل في كل الحالات. وجود مرجع مستقل يحمي المقارنة من التفسير الخاطئ.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts. Cloud credentials must not be committed, and personal health data do not belong in the educational example.

PYTHON

```python
from pyscf import gto, scf, fci

def h2_fci():
    mol = gto.M(
        atom="H 0 0 0; H 0 0 0.735",
        basis="sto3g",
        unit="Angstrom",
        charge=0,
        spin=0,
    )

    mf = scf.RHF(mol).run()
    solver = fci.FCI(mol, mf.mo_coeff)

    energy, vector = solver.kernel(
        mf.mo_coeff,
        mol.nao_nr(),
        mol.nelectron,
    )

    return {
        "hf_energy": float(mf.e_tot),
        "fci_energy": float(energy),
        "state_dimension": int(len(vector)),
    }

if __name__ == "__main__":
    print(h2_fci())
```

ملف المصدر: src/quantum_molecules.py — تنفيذ H_2O وCO_2 بفضاء نشط

Source File: src/quantum_molecules.py — ActiveSpaceTransformer makes H_2O and CO_2 executable without claiming a complete all-electron representation.

ActiveSpaceTransformer يجعل H_2O وCO_2 قابلين للتجربة دون الادعاء أنهما تمثيل كامل لكل الإلكترونات.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts. Cloud credentials must not be committed, and personal health data do not belong in the educational example.

PYTHON

```python
from qiskit_nature.units import DistanceUnit
from qiskit_nature.second_q.drivers import PySCFDriver
from qiskit_nature.second_q.transformers import ActiveSpaceTransformer

MOLECULES = {
    "H2O": (
        "O 0 0 0; "
        "H 0.758 0 0.504; "
        "H -0.758 0 0.504"
    ),
    "CO2": (
        "O 0 0 0; "
        "C 0 0 1.16; "
        "O 0 0 2.32"
    ),
}

def build_active_problem(name):
    full = PySCFDriver(
        atom=MOLECULES[name],
        basis="sto3g",
        unit=DistanceUnit.ANGSTROM,
    ).run()

    return ActiveSpaceTransformer(
        num_electrons=2,
        num_spatial_orbitals=2,
    ).transform(full)

if __name__ == "__main__":
    for name in MOLECULES:
        p = build_active_problem(name)
        print(
            name,
```

```
        "particles=", p.num_particles,
        "orbitals=", p.num_spatial_orbitals,
    )
```

ملف المصدر: src/quantum_h2.py — تنفيذ H_2 مع مرجع كلاسيكي

Source File: src/quantum_h2.py — The quantum workflow keeps a classical reference and environment record so an energy value never becomes a context-free number.

المسار الكمومي يحتفظ بمرجع كلاسيكي وبسجل للبيئة حتى لا تصبح قيمة الطاقة رقمًا بلا سياق.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts. Cloud credentials must not be committed, and personal health data do not belong in the educational example.

PYTHON

```python
from qiskit_nature.units import DistanceUnit
from qiskit_nature.second_q.drivers import PySCFDriver
from qiskit_nature.second_q.mappers import ParityMapper
from qiskit_nature.second_q.circuit.library import HartreeFock, UCCSD
from qiskit_nature.second_q.algorithms import GroundStateEigensolver
from qiskit_algorithms import VQE
from qiskit_algorithms.optimizers import L_BFGS_B
from qiskit.primitives import StatevectorEstimator

def run_h2():
    problem = PySCFDriver(
        atom="H 0 0 0; H 0 0 0.735",
        basis="sto3g",
        unit=DistanceUnit.ANGSTROM,
    ).run()

    mapper = ParityMapper(
        num_particles=problem.num_particles
    )

    ansatz = UCCSD(
        problem.num_spatial_orbitals,
        problem.num_particles,
        mapper,
        initial_state=HartreeFock(
            problem.num_spatial_orbitals,
            problem.num_particles,
            mapper,
        ),
    )

    vqe = VQE(
```

```
        StatevectorEstimator(),
        ansatz,
        L_BFGS_B(),
    )
    vqe.initial_point = [0.0] * ansatz.num_parameters

    solver = GroundStateEigensolver(mapper, vqe)
    return solver.solve(problem)

if __name__ == "__main__":
    print(run_h2())
```

ملف المصدر: src/calibration.py — معايرة احتمالات النموذج

Source File: src/calibration.py — When probabilities are part of a report, their calibration should be evaluated separately from accuracy.

عندما تكون الاحتمالات جزءًا من التقرير يجب اختبار معايرتها بشكل مستقل عن الدقة.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts. Cloud credentials must not be committed, and personal health data do not belong in the educational example.

PYTHON

```
from sklearn.calibration import (
    CalibratedClassifierCV,
    calibration_curve,
)
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import brier_score_loss

def calibrated_model():
    base = LogisticRegression(max_iter=2000)
    return CalibratedClassifierCV(
        base, method="sigmoid", cv=5
    )

def calibration_report(model, X, y):
    p = model.predict_proba(X)[:, 1]
    frac, mean_pred = calibration_curve(
        y, p, n_bins=5, strategy="uniform"
    )
    return {
        "brier": float(brier_score_loss(y, p)),
        "fraction_positive": frac.tolist(),
        "mean_predicted": mean_pred.tolist(),
    }
```

ملف المصدر: src/ml_pipeline.py — خط تعلم آلي قابل لإعادة الإنتاج

Source File: src/ml_pipeline.py — This pipeline joins preprocessing, splitting, and evaluation while preventing test information from leaking into training.

هذا المسار يربط المعالجة والتقسيم والتقييم ويمنع تسرب معلومات الاختبار إلى التدريب.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts. Cloud credentials must not be committed, and personal health data do not belong in the educational example.

PYTHON

```python
from dataclasses import dataclass
from sklearn.model_selection import cross_validate, StratifiedKFold
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression

@dataclass(frozen=True)
class Evaluation:
    accuracy_mean: float
    accuracy_std: float
    fit_time_mean: float

def make_model():
    return Pipeline([
        ("imputer", SimpleImputer(strategy="median")),
        ("scale", StandardScaler()),
        ("classifier", LogisticRegression(max_iter=2000)),
    ])

def evaluate(X, y):
    cv = StratifiedKFold(
        n_splits=5, shuffle=True, random_state=42
    )

    result = cross_validate(
        make_model(),
        X,
        y,
        cv=cv,
        scoring="accuracy",
        return_train_score=False,
    )

    return Evaluation(
        accuracy_mean=float(result["test_score"].mean()),
        accuracy_std=float(result["test_score"].std()),
```

```
        fit_time_mean=float(result["fit_time"].mean()),
    )
```

ملف المصدر: src/features.py — بناء الميزات مع المصدر والإصدار

Source File: src/features.py — A feature is not a number without context. The feature matrix should carry source and version so model changes can be traced.

الميزة ليست رقمًا بلا سياق. يجب أن تحمل مصفوفة الميزات مصدرها وإصدارها حتى يمكن تفسير تغير النموذج.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts. Cloud credentials must not be committed, and personal health data do not belong in the educational example.

PYTHON

```python
from dataclasses import dataclass
import numpy as np

@dataclass(frozen=True)
class FeatureSet:
    values: np.ndarray
    feature_names: tuple[str, ...]
    data_version: str
    source: str

def build_features(elements):
    values = np.array(
        [[
            e.atomic_number,
            e.atomic_weight,
            e.period,
            e.group_number or 0,
        ] for e in elements],
        dtype=float,
    )

    names = (
        "atomic_number",
        "atomic_weight",
        "period",
        "group_number",
    )

    return FeatureSet(
        values=values,
        feature_names=names,
```

```
        data_version="elements-v1",
        source="project.db",
    )
```

ملف المصدر: src/repository.py — نمط المستودع بين OOP وSQL

Source File: src/repository.py — The repository lets the application work with objects instead of scattering SQL. It is a testable boundary that can later be replaced.

المستودع يجعل التطبيق يتعامل مع كائنات بدل نشر SQL في كل طبقة. وهو حد يمكن اختباره وتغييره.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts. Cloud credentials must not be committed, and personal health data do not belong in the educational example.

PYTHON

```python
import sqlite3
from .models import Element

class ElementRepository:
    def __init__(self, connection: sqlite3.Connection):
        self.con = connection

    def get(self, atomic_number: int):
        row = self.con.execute(
            "SELECT atomic_number,symbol,name,atomic_weight,"
            "category,period,group_number FROM elements "
            "WHERE atomic_number = ?",
            (atomic_number,),
        ).fetchone()

        if row is None:
            return None
        return Element(*row)

    def count(self) -> int:
        return self.con.execute(
            "SELECT COUNT(*) FROM elements"
        ).fetchone()[0]

    def all(self):
        rows = self.con.execute(
            "SELECT atomic_number,symbol,name,atomic_weight,"
            "category,period,group_number FROM elements "
            "ORDER BY atomic_number"
        )
        return [Element(*row) for row in rows]
```

ملف المصدر: src/models.py — نماذج OOP للكيانات والنتائج

Source File: src/models.py — The OOP layer separates scientific entities from persistence. Element represents an element record, Experiment an execution, and Result a measurement.

طبقة OOP تفصل الكيان العلمي عن طريقة التخزين. Element يمثل سجل العنصر، وExperiment يصف تشغيلًا، وResult يصف قياسًا.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts. Cloud credentials must not be committed, and personal health data do not belong in the educational example.

PYTHON

```python
from dataclasses import dataclass, asdict
from typing import Optional

@dataclass(frozen=True)
class Element:
    atomic_number: int
    symbol: str
    name: str
    atomic_weight: float
    category: str
    period: int
    group_number: Optional[int] = None

    def as_record(self):
        return asdict(self)

@dataclass(frozen=True)
class Experiment:
    experiment_id: str
    experiment_type: str
    data_version: str
    software_version: str
    created_at: str
    status: str = "pending"

@dataclass(frozen=True)
class Result:
    experiment_id: str
    metric_name: str
    metric_value: Optional[float]
    unit: Optional[str]
    provenance: str

def validate_element(e: Element) -> None:
```

```
    if not 1 <= e.atomic_number <= 118:
        raise ValueError("atomic_number must be in 1..118")
    if not e.symbol:
        raise ValueError("symbol is required")
    if e.atomic_weight <= 0:
        raise ValueError("atomic_weight must be positive")
```

ملف المصدر: scripts/load_elements.py — تحميل العناصر والتحقق من اكتمال السجل

Source File: scripts/load_elements.py — The loader does not assume that a CSV file is correct merely because it exists. It validates atomic numbers, duplicates, and positive weights, then uses one transaction.

التحميل لا يفترض أن ملف CSV صحيح لمجرد أنه موجود. يتحقق من العدد الذري وعدم التكرار والأوزان الموجبة ثم يستخدم معاملة واحدة.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts. Cloud credentials must not be committed, and personal health data do not belong in the educational example.

PYTHON

```
from pathlib import Path
import csv
import sqlite3

DB = Path("data/project.db")
CSV_PATH = Path("data/elements.csv")

def validate(row):
    z = int(row["atomic_number"])
    weight = float(row["atomic_weight"])
    if not 1 <= z <= 118:
        raise ValueError(f"invalid atomic number: {z}")
    if weight <= 0:
        raise ValueError(f"invalid atomic weight: {weight}")

def load():
    con = sqlite3.connect(DB)
    con.execute("PRAGMA foreign_keys = ON")
    try:
        with CSV_PATH.open(encoding="utf-8", newline="") as fh:
            rows = list(csv.DictReader(fh))

        if len(rows) != 118:
            raise ValueError(
                f"expected 118 elements, found {len(rows)}"
```

```
        )

      for row in rows:
        validate(row)

      con.executemany(
        "INSERT INTO elements "
        "(atomic_number,symbol,name,atomic_weight,"
        "category,period,group_number) "
        "VALUES (?,?,?,?,?,?,?) "
        "ON CONFLICT(atomic_number) DO UPDATE SET "
        "symbol=excluded.symbol,name=excluded.name,"
        "atomic_weight=excluded.atomic_weight,"
        "category=excluded.category,period=excluded.period,"
        "group_number=excluded.group_number",
        [
          (
            int(r["atomic_number"]), r["symbol"], r["name"],
            float(r["atomic_weight"]), r["category"],
            int(r["period"]),
            int(r["group_number"]) if r["group_number"] else None,
          )
          for r in rows
        ],
      )
      con.commit()
    except Exception:
      con.rollback()
      raise
    finally:
      con.close()

if __name__ == "__main__":
    load()
    print("118-element load: PASS")
```

ملف المصدر: scripts/schema.sql — مخطط قاعدة البيانات الأساسي

Source File: scripts/schema.sql — This file defines the tables for element identity, experiments, results, and audit events. Keys, constraints, and indexes are part of the scientific contract.

هذا الملف يعرّف الجداول التي تحمل هوية العناصر والتجارب والنتائج وسجل التدقيق. القيود والمفاتيح والفهارس جزء من العقد العلمي.

A publication-ready project should make its system files understandable. The database is separated from scripts, tests, reports, and synthetic data. run_pipeline.py coordinates stages, requirements.txt pins the environment, and tests encode contracts. Cloud credentials must not be committed, and personal health data do not belong in the educational example.

SQL

```sql
PRAGMA foreign_keys = ON;

CREATE TABLE IF NOT EXISTS elements (
    atomic_number INTEGER PRIMARY KEY,
    symbol TEXT NOT NULL UNIQUE,
    name TEXT NOT NULL,
    atomic_weight REAL NOT NULL CHECK (atomic_weight > 0),
    category TEXT NOT NULL,
    period INTEGER NOT NULL CHECK (period BETWEEN 1 AND 7),
    group_number INTEGER
);

CREATE TABLE IF NOT EXISTS experiments (
    experiment_id TEXT PRIMARY KEY,
    experiment_type TEXT NOT NULL,
    data_version TEXT NOT NULL,
    software_version TEXT NOT NULL,
    created_at TEXT NOT NULL,
    status TEXT NOT NULL CHECK (
        status IN ('pending','running','completed','failed')
    )
);

CREATE TABLE IF NOT EXISTS results (
    result_id INTEGER PRIMARY KEY,
    experiment_id TEXT NOT NULL,
    metric_name TEXT NOT NULL,
    metric_value REAL,
    unit TEXT,
    provenance TEXT NOT NULL,
    FOREIGN KEY (experiment_id) REFERENCES experiments(experiment_id)
);

CREATE TABLE IF NOT EXISTS audit_event (
    event_id INTEGER PRIMARY KEY,
    request_id TEXT NOT NULL,
    event_type TEXT NOT NULL,
    actor TEXT NOT NULL,
    created_at TEXT NOT NULL,
    payload_json TEXT NOT NULL
);

CREATE INDEX IF NOT EXISTS idx_results_experiment
ON results(experiment_id);

CREATE INDEX IF NOT EXISTS idx_audit_request
ON audit_event(request_id);
```

PYTHON

```python
from dataclasses import dataclass

@dataclass(frozen=True)
class Failure:
    run_id: str
    stage: str
    reason: str
    status: str = "failed"

failure = Failure(
    run_id="run-002",
    stage="quantum",
    reason="backend unavailable",
)
```

```
print(failure)
```

إذا تعذر تثبيت مكتبة أو فشل اتصال بخدمة كمومية، فلا ينبغي كتابة رقم افتراضي في التقرير. يسجل النظام الحالة failed والمرحلة والسبب والنسخة والمدخلات التي كانت متاحة. هذا مهم في مشروع قابل لإعادة الإنتاج لأن الفشل قد يكون سببه بيئة أو خدمة وليس الخوارزمية نفسها. عند إعادة التشغيل يجب أن يكون run_id جديدًا، وأن تبقى محاولة الفشل السابقة محفوظة. هكذا يستطيع المراجع معرفة أن النتيجة الحالية جاءت من تشغيل ناجح بعد فشل سابق، لا من سجل تم تنظيفه.

If a dependency cannot be installed or a quantum service call fails, the report should not contain a fabricated number. The system records failed status, stage, reason, version, and available inputs. This matters in reproducible projects because a failure may be environmental or service-related rather than algorithmic. A rerun should receive a new run_id while the failed attempt remains stored. An auditor can then see that the current result came from a successful rerun after a failure, rather than from a cleaned-up history.

مذكرة مختبر: فشل التشغيل جزء من النتيجة

Lab Note: Execution Failure Is a Result

PYTHON

```python
import numpy as np

A = np.diag([1.0, 3.0])
b = np.array([1.0, 0.5])

x = np.linalg.solve(A, b)
residual = np.linalg.norm(A @ x - b)

print("x:", x)
print("residual:", residual)
```

القيمة التعليمية لـHHL تأتي من وضوح شروطه. نحتاج مصفوفة مناسبة يمكن تمثيلها كمؤثر كمومي، وحالة إدخال يمكن تحضيرها، وآلية لتقدير القيم الذاتية، ودوران مشروط، ثم استخراج كمية مفيدة من الحالة الناتجة. كذلك يؤثر عدد الشرط في صعوبة الحل واستخراج المعلومات. لذلك لا يصح أن نكتب «HHL يحل شرودنجر» ثم نتوقف. يجب أن نحدد كيف تحولت المسألة الفيزيائية إلى Ax=b وما الذي

نقيسه من x. في المثال الصغير يمكن حل النظام كلاسيكيًا أولًا ثم مقارنة البنية الكمومية التعليمية بالمرجع.

The educational value of HHL lies in making its assumptions explicit. We need a matrix suitable for quantum representation, a preparable input state, a mechanism for eigenvalue estimation, a controlled rotation, and a measurable quantity extracted from the resulting state. The condition number also affects difficulty and information extraction. It is therefore not correct to write “HHL solves Schrödinger” and stop. A physical problem must first be transformed into Ax=b, and the measured quantity from x must be defined. For a small example, solve the system classically first and compare the educational quantum construction with that reference.

مذكرة مختبر: HHL وشروط المسألة الخطية

Lab Note: HHL and the Conditions of the Linear System

PYTHON

```
import math

def binomial_standard_error(p, shots):
    return math.sqrt(p * (1 - p) / shots)

for shots in (100, 1000, 10000):
    print(
        shots,
        binomial_standard_error(0.5, shots)
    )
```

عدد shots ليس مجرد خيار تجميلي. القياس الكمومي يعطي عينات، وزيادة عددها تقلل عدم اليقين الإحصائي عادةً، لكنها لا تزيل أخطاء الجهاز المنهجية. لذلك يجب الفصل بين statistical sampling error وبين hardware noise. عند مقارنة جهاز بمحاكي مثالي، ينبغي عرض الاثنين مع عدد shots نفسه إن كان ذلك ممكنًا، أو توضيح سبب اختلافه. وإذا استخدمنا تخفيف خطأ، يجب تسجيل الإعداد بدل عرض النتيجة وكأنها قياس خام.

Shot count is not a cosmetic option. Quantum measurement produces samples, and increasing the number of shots generally

reduces sampling uncertainty, but it does not remove systematic hardware errors. Statistical sampling error and hardware noise should therefore be distinguished. When comparing hardware with an ideal simulator, use the same shot count when practical or explain why it differs. If error mitigation is used, record the settings rather than presenting the output as a raw measurement.

مذكرة مختبر: الضوضاء واللقطات

Lab Note: Noise and Shots

PYTHON

```
from qiskit import QuantumCircuit, transpile

qc = QuantumCircuit(2)
qc.h(0)
qc.cx(0, 1)

compiled = transpile(qc, basis_gates=["rz", "sx", "x", "cx"])

print(compiled)
print("depth:", compiled.depth())
print("size:", compiled.size())
```

عند استخدام خدمة كمومية حقيقية، يجب حفظ نسخة الدائرة التي أرسلت إلى الخلفية بعد التحويل إلى تعليمات الهدف. واجهات Runtime الحديثة تتطلب دوائر ومرصودات متوافقة مع الهدف قبل التنفيذ. لذلك يجب أن يظهر في سجل التجربة backend وprimitive والـPUB والـshots أو precision ووقت الإرسال وحالة المهمة. هذا يجعل النتيجة قابلة للمقارنة مع تشغيل لاحق على خلفية أخرى. إذا تغيرت الدائرة بسبب transpilation، فإن تغييرها جزء من التجربة وليس تفصيلًا يمكن تجاهله.

When using real quantum hardware, the circuit sent to the backend should be recorded after it has been transformed into target-compatible instructions. Modern Runtime interfaces require circuits and observables to conform to the target before execution. The experiment record should therefore include backend, primitive, PUB definition, shots or precision, submission time, and job status. This makes results comparable across runs on different backends. If

transpilation changes the circuit, that change is part of the experiment rather than an irrelevant detail.

مذكرة مختبر: Qiskit Runtime وتسجيل الدائرة

Lab Note: Qiskit Runtime and Circuit Recording

PYTHON

```
ALLOWED_FIELDS = {
    "symbol", "name", "atomic_weight",
    "category", "period", "group_number",
}

def validate_query(field):
    if field not in ALLOWED_FIELDS:
        raise ValueError("field is not allowed")

validate_query("atomic_weight")
print("query contract: PASS")
```

يمكن للنموذج اللغوي أن يساعد في إنشاء استعلام SQL أو تلخيص نتيجة، لكن يجب أن يمر المخرج عبر مخطط وقواعد تحقق. إذا طلب المستخدم سؤالًا عن عنصر، يمكن للنظام تحويله إلى بنية مثل entity وfield وfilter ثم تنفيذ SQL بنفسه. لا ينبغي للنموذج أن يخترع قيمة atomic_weight أو نتيجة VQE. إذا لم يجد مصدرًا، تكون الإجابة «لا يوجد مصدر في السجل» بدل ملء الفراغ بتخمين. هذا الفصل بين التوليد والتنفيذ يقلل خطر أن تتحول اللغة الطبيعية إلى مصدر بيانات غير موثق.

An LLM can help construct an SQL query or summarize a result, but its output should pass a schema and validation rules. A user question about an element can be converted into fields such as entity, field, and filter, after which the system executes SQL itself. The model should not invent an atomic weight or VQE result. If the database has no source, the system should report that no source is present rather than filling the gap with a guess. Separating generation from execution reduces the risk of natural language becoming an undocumented data source.

مذكرة مختبر: LLM كسكرتير بيانات لا كمرجع علمي

Lab Note: An LLM as a Data Clerk, Not a Scientific Reference

PYTHON

```
def score_candidates(candidates):
    valid = 0
    for item in candidates:
        if 0 < item["atomic_weight"] < 400:
            valid += 1
    return valid / max(len(candidates), 1)

print(score_candidates([
    {"atomic_weight": 1.008},
    {"atomic_weight": 300.0},
    {"atomic_weight": -4.0},
]))
```

المولد لا يقاس بعدد العينات التي أنتجها. يجب أن نختبر صلاحية البنية، وتوزيع الخصائص، والتنوع، وإعادة الإنتاج، ونسبة العينات التي تفشل القيود. في مشروع العناصر يمكن تعريف حدود فيزيائية بسيطة ثم رفض القيم الخارجة. إذا كان النموذج يولد عينات متشابهة جدًا، فالتنوع ضعيف. وإذا كان يولد قيمًا خارج النطاق بكثرة، فهناك مشكلة في التدريب أو تمثيل البيانات. هذه الاختبارات تحول المولد من عرض بصري إلى نظام قابل للقياس.

A generator is not evaluated by how many samples it produces. We should test structural validity, feature distributions, diversity, reproducibility, and the fraction of candidates that violate constraints. For element data, simple physical ranges can be defined and invalid values rejected. If outputs are nearly identical, diversity is weak; if many values fall outside valid ranges, the model or representation needs review. These tests turn generation from a visual demonstration into a measurable system.

مذكرة مختبر: تقييم النموذج التوليدي

Lab Note: Evaluating a Generative Model

PYTHON

```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_validate
```

```
model = Pipeline([
    ("scale", StandardScaler()),
    ("clf", LogisticRegression(max_iter=2000)),
])

scores = cross_validate(
    model, X, y, cv=5, scoring="accuracy"
)

print(scores["test_score"])
```

البيانات الصغيرة تجعل تسرب البيانات أخطر لأن كل صف قد يؤثر في التقييم بدرجة كبيرة. يجب أن تتم عملية التقسيم قبل أي عملية تعلم للإحصاءات مثل المتوسط والانحراف المعياري. Pipeline في scikit-learn يساعد على ربط هذه العمليات بالتقسيم، لكن يجب أيضًا فحص مصدر كل ميزة. إذا كانت الميزة مشتقة من النتيجة التي نحاول التنبؤ بها، فحتى أفضل Pipeline لن ينقذ التصميم. لذلك يراجع المشروع معنى الميزة قبل شكلها العددي.

Small datasets make leakage especially dangerous because each row can strongly affect evaluation. Splitting should happen before learning statistics such as means and standard deviations. A scikit-learn Pipeline helps bind preprocessing to the split, but feature provenance must also be inspected. If a feature is derived from the target, even the best Pipeline cannot rescue the design. The project therefore reviews the semantic meaning of a feature before its numerical representation.

مذكرة مختبر: منع تسرب البيانات في التجارب الصغيرة

Lab Note: Preventing Leakage in Small Experiments

PYTHON

```
import json
import numpy as np

features = np.asarray(X, dtype=float)
metadata = {
    "feature_names": list(feature_names),
    "data_version": "elements-v1",
}

np.save("artifacts/features.npy", features)

with open(
    "artifacts/features.json",
    "w",
    encoding="utf-8",
```

```
) as fh:
    json.dump(metadata, fh, ensure_ascii=False, indent=2)
```

الانتقال من SQL إلى NumPy قد يخلق خطأ خطيرًا إذا تغير ترتيب الأعمدة. النموذج لا يعرف أن العمود الثاني كان atomic_weight وأنه أصبح period. لذلك يجب حفظ feature_names مع المصفوفة، ويفضل حفظ إصدار بناء الميزات. عند إعادة التدريب يجب مقارنة قائمة الأسماء والأنواع والوحدات. هذا العقد مهم أيضًا عند استخدام Dask أو Spark؛ تقسيم الصفوف لا ينبغي أن يغير معنى الأعمدة. إذا كانت المصفوفة محفوظة بصيغة binary، يجب حفظ metadata منفصلة أو ضمن حاوية تدعمها.

Moving from SQL to NumPy can create a dangerous error if column order changes. A model does not know that the second column used to be atomic_weight and has become period. Store feature_names with the matrix and preferably a feature-build version. On retraining, compare names, types, and units. The same contract matters with Dask or Spark; row partitioning must not change column semantics. If a binary matrix is stored, its metadata should be stored alongside it or in a format that supports schema information.

مذكرة مختبر: مصفوفة الميزات وعقد ترتيب الأعمدة

Lab Note: Feature Matrices and Column-Order Contracts

SQL

```sql
SELECT g.gene_id, v.variant_id, e.evidence_level
FROM genes AS g
JOIN variants AS v
  ON v.gene_id = g.gene_id
JOIN evidence AS e
  ON e.variant_id = v.variant_id
WHERE g.gene_id = ?;
```

في مشروع Gene Resistance Tracker، العلاقة بين gene وvariant وevidence ليست مجرد ربط أسماء. يجب أن يوضح المفتاح ما إذا كانت العلاقة واحدًا إلى واحد أو واحدًا إلى كثير أو كثيرًا إلى كثير. عند وجود many-to-many نستخدم جدول ربط يحمل خصائص العلاقة نفسها، مثل evidence_level وconfidence وsource_version. هذا يمنع فقدان سياق الدليل عند

ربط الكيانات. كما يجب اختبار JOIN على بيانات صغيرة معروفة قبل تشغيله على ملايين السجلات. إذا أعاد JOIN صفوفًا أكثر من المتوقع، قد يكون هناك تكرار دلالي وليس مجرد خطأ في SQL.

In Gene Resistance Tracker, the relationship among genes, variants, and evidence is more than a name match. The schema should express whether a relationship is one-to-one, one-to-many, or many-to-many. A many-to-many relationship uses a junction table that stores properties of the relationship itself, such as evidence_level, confidence, and source_version. This preserves the context of evidence during joins. JOIN logic should be tested on a small known dataset before being run on millions of records. If a join returns more rows than expected, the problem may be semantic duplication rather than a syntax error.

مذكرة مختبر: JOIN ليس مجرد عملية ربط

Lab Note: JOIN Is More Than a Link

SQL

```
EXPLAIN QUERY PLAN
SELECT sample_id, position, ref, alt
FROM genomic_observation
WHERE chromosome = ?
  AND position BETWEEN ? AND ?
ORDER BY position;

CREATE INDEX IF NOT EXISTS idx_genomic_region
ON genomic_observation(chromosome, position);
```

عندما يكبر الحجم لا يكفي أن يعمل الاستعلام؛ يجب أن نعرف لماذا يعمل بالسرعة التي يعمل بها. تبدأ المراجعة بخطة التنفيذ ثم الفهرس المناسب ثم قياس القراءة والكتابة. قد يسرّع الفهرس استعلامًا واحدًا لكنه يزيد كلفة INSERT وUPDATE ويستهلك مساحة. لذلك يجب اختبار السيناريو الفعلي بدل إنشاء فهرس لكل عمود. في البيانات الجينومية تكون الأعمدة مثل sample_id وchromosome وposition مرشحة للاستعلام، لكن الفائدة تعتمد على نمط الوصول. كما ينبغي تجنب الاعتماد على ترتيب الصفوف الضمني؛ ترتيب النتائج يجب أن يكون صريحًا باستخدام ORDER BY.

When data volume grows, it is not enough that a query works; we should know why it performs as it does. Review the execution plan,

select an index based on the workload, and measure read and write behavior. An index can speed one query while increasing INSERT and UPDATE costs and consuming storage. Test the actual workload instead of indexing every column. In genomic-style data, sample_id, chromosome, and position may be useful query keys, but their value depends on access patterns. Do not rely on implicit row order; result order should be explicit through ORDER BY.

مذكرة مختبر: SQL تحت ضغط البيانات

Lab Note: SQL Under Data Pressure

PYTHON

```
import numpy as np
import pandas as pd

rng = np.random.default_rng(42)
rows = 1_000_000

df = pd.DataFrame({
    "sample_id": rng.integers(1, 100_001, rows),
    "chromosome": rng.integers(1, 23, rows),
    "position": rng.integers(1, 250_000_001, rows),
    "ref": rng.choice(list("ACGT"), rows),
    "alt": rng.choice(list("ACGT"), rows),
})

print("synthetic rows:", len(df))
```

البيانات الاصطناعية هنا ليست بديلًا عن دراسة جينومية حقيقية. وظيفتها أن تجعل مشكلة الحجم قابلة للتجربة دون إدخال معلومات صحية شخصية. يمكن إنشاء ملايين السجلات التي تحتوي على sample_id وموقع كروموسومي ومتغير، ثم اختبار الذاكرة وسرعة الكتابة والقراءة وعمليات JOIN. المهم أن يحمل كل ملف seed معروفًا حتى يمكن إعادة توليد البيانات نفسها. كما يجب أن يعلن التقرير بوضوح أنها بيانات اصطناعية. عندما ينجح النظام على هذا المثال، لا يجوز القفز إلى استنتاج أن النظام جاهز تلقائيًا لبيانات سريرية؛ البيانات الحقيقية تضيف متطلبات الخصوصية والجودة والتحقق والحوكمة.

The synthetic dataset is not a substitute for a real genomic study. Its role is to make the scale problem testable without introducing personal health information. Millions of records can contain

sample_id, genomic position, and variant fields, allowing memory, read/write, and JOIN behavior to be measured. The generator should use a known seed so the same dataset can be reproduced. The report must explicitly identify the data as synthetic. Successful execution on this benchmark does not mean that the system is automatically ready for clinical data; real data add privacy, quality, validation, and governance requirements.

مذكرة مختبر: تصميم البيانات الجينومية الاصطناعية

Lab Note: Designing Synthetic Genomic Data

PYTHON

```python
def compare(reference, candidate):
    return {
        "reference": float(reference),
        "candidate": float(candidate),
        "absolute_error": abs(
            float(candidate) - float(reference)
        ),
    }

print(compare(-1.137306, -1.136900))
```

كل نتيجة مهمة تحتاج إلى مرجع محدد قبل تشغيل الخوارزمية. في H_2 يمكن أن يكون المرجع Hartree–Fock أو FCI، وفي التعلم الآلي نموذجًا بسيطًا، وفي المعالجة الموزعة تنفيذًا محليًا. تحديد المرجع مسبقًا يمنع اختيار مقارنة مريحة بعد ظهور النتيجة. كما يجب تحديد مقياس المقارنة: خطأ مطلق، خطأ نسبي، فرق في الطاقة، دقة، معايرة، أو زمن تنفيذ. إذا تحسنت النتيجة في مقياس وتدهورت في آخر، يجب عرض الاثنين بدل اختيار الرقم الأفضل فقط. المرجع ليس حكمًا مطلقًا، لكنه نقطة ثابتة تساعد القارئ على فهم مقدار التغيير.

Every important result needs a defined reference before the algorithm is executed. For H_2 the reference can be Hartree–Fock or FCI; for machine learning it can be a simple baseline; for distributed processing it can be a local implementation. Defining the reference in advance prevents a convenient comparison from being selected after seeing the result. The comparison metric should also be declared: absolute error, relative error, energy difference, accuracy,

calibration, or execution time. If one metric improves while another worsens, both should be reported rather than selecting only the favorable number. A reference is not an absolute judgment, but it gives the reader a stable point for interpreting change.

مذكرة مختبر: المرجع قبل الخوارزمية

Lab Note: The Reference Before the Algorithm

PYTHON

```
expected = set(range(1, 119))
observed = set(df["atomic_number"].astype(int))

assert observed == expected
assert df["symbol"].is_unique
assert len(df) == 118

print("identity contract: PASS")
```

التحقق من الجدول الدوري يبدأ من الهوية لا من التحليل. يجب أن تكون الأعداد الذرية مجموعة كاملة من 1 إلى 118، وأن يكون الرمز فريدًا، وأن يرتبط كل سجل باسم عنصر واحد. هذا الاختبار البسيط يمنع خطأً صامتًا قد ينتقل إلى كل المراحل اللاحقة. إذا فقد عنصر أو تكرر، يمكن أن تتغير المتوسطات والمصفوفات ونتائج التصنيف دون أن ينهار البرنامج. لذلك يجب أن يسجل التشغيل عدد الصفوف، والحد الأدنى والأقصى للعدد الذري، وعدد الرموز الفريدة، ثم يقارنها بالمجموعة المتوقعة. لا يعني نجاح هذا الاختبار أن البيانات العلمية صحيحة في كل تفاصيلها؛ إنه يثبت فقط أن عقد الهوية الأساسي قد اجتاز الفحص. بعد ذلك يمكن الانتقال إلى الوحدات والمصادر والقيم المرجعية.

Periodic-table validation begins with identity rather than analysis. Atomic numbers should form the complete set from 1 through 118, symbols should be unique, and each record should map to one element name. This simple test prevents a silent error from propagating through later stages. If one element is missing or duplicated, averages, matrices, and classification outputs can change without the program crashing. The run should therefore record row count, minimum and maximum atomic number, and the number of unique symbols, then compare them with the expected

set. Passing this test does not prove that every scientific field is correct; it only proves that the primary identity contract passed. Source validation, units, and reference values are separate checks.

مذكرة مختبر: عقد الهوية للجدول الدوري

Lab Note: The Periodic-Table Identity Contract

PYTHON

```
from pathlib import Path

text = Path(
    "final/Brick_by_Brick_Part_III.txt"
).read_text(encoding="utf-8")

assert "\uFFFD" not in text
for term in [
    "Chapter 33:",
    "Chapter 38:",
    "H₂O",
    "CO₂",
    "HHL",
]:
    assert term in text

print("publication text contract: PASS")
```

.

مذكرة مختبر: اختبار النص والفهرس قبل النشر

Lab Note: Text and TOC Validation Before Publication

PYTHON

```
from datetime import datetime, timezone

record = {
    "backend": "recorded-at-runtime",
    "primitive": "EstimatorV2",
    "shots": 4096,
    "submitted_at": datetime.now(
        timezone.utc
    ).isoformat(),
}

print(record)
```

تشغيل دائرة على جهاز حقيقي يجب أن يسجل backend name وprimitive والإصدار والخيارات وعدد shots والوقت. إذا استخدمنا IBM Quantum أو خدمة أخرى، فإن اسم الخدمة

وحده لا يكفي لإعادة التجربة لأن حالة الجهاز تتغير. لذلك تحفظ بيانات الخلفية التي يسمح بها النظام، ويضاف job identifier إلى سجل التجربة. لا تُحفظ أسرار الاعتماد داخل الكتاب أو المستودع.

Running a circuit on real hardware should record backend name, primitive, software version, options, shots, and time. A service name alone is not enough to reproduce a run because hardware state changes. Store permitted backend metadata and the job identifier in the experiment record. Credentials must never be stored in the book or repository.

مذكرة مختبر: سجل الأجهزة والخدمات الكمومية

Lab Note: Quantum Backend and Service Records

PYTHON

```
import dask.dataframe as dd
import time

ddf = dd.read_parquet(
    "data/synthetic_genomics/*.parquet"
)

start = time.perf_counter()
rows = ddf.shape[0].compute()
elapsed = time.perf_counter() - start

print("rows:", rows)
print("seconds:", elapsed)
print("partitions:", ddf.npartitions)
```

التوسع ليس عدد العمال فقط. يجب مراقبة الذاكرة لأن جزءًا كبيرًا جدًا قد يسبب spilling أو فشل العامل، بينما الأجزاء الصغيرة جدًا ترفع عدد المهام. في اختبار تعليمي يمكن قياس حجم كل ملف وعدد الصفوف في كل partition ثم مقارنة زمن compute. لا نحتاج إلى تشغيل Cluster حقيقي على 118 سجلًا؛ الجدول الدوري يشرح API، بينما البيانات الاصطناعية الجينومية تبرر القياس.

Scaling is not only the number of workers. Memory must be monitored because very large partitions can cause spilling or worker failure, while tiny partitions increase task overhead. A teaching benchmark can record file size, rows per partition, and compute time. There is no need for a real cluster for 118 element records; the

periodic table explains the API, while synthetic genomic data justify scale testing.

مذكرة مختبر: مراقبة الذاكرة في Dask

Lab Note: Memory Monitoring in Dask

PYTHON

```
from sklearn.neighbors import NearestNeighbors
from sklearn.preprocessing import StandardScaler

X_scaled = StandardScaler().fit_transform(X)
model = NearestNeighbors(
    n_neighbors=5, metric="euclidean"
).fit(X_scaled)

distances, indices = model.kneighbors(
    X_scaled[:1]
)

print(distances[0])
print(indices[0])
```

إذا استخدم المشروع NearestNeighbors أو مسافة إقليدية، فالنتيجة تعني أن السجل قريب من سجلات أخرى وفق الميزات المختارة فقط. لا تعني أنه أفضل مادة أو أنه مناسب لتطبيق طبي. يجب عرض المسافة والميزات المستخدمة، وربما مجموعة المرشحين، حتى يستطيع القارئ فهم سبب التوصية. كما يجب اختبار حساسية النتيجة لتغيير مقياس الميزات؛ إذا تغير المرشح الأول بشدة بعد إعادة التقييس، فالتوصية غير مستقرة.

If the project uses NearestNeighbors or Euclidean distance, the result means that one record is close to others under the selected features. It does not mean that it is the best material or suitable for a medical application. Show distance and selected features so the reader can understand the recommendation. Test sensitivity to feature scaling; if the top candidate changes substantially after rescaling, the recommendation is unstable.

مذكرة مختبر: حدود التوصية القائمة على التشابه

Lab Note: Limits of Similarity-Based Recommendation

PYTHON

```
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

X_scaled = StandardScaler().fit_transform(X)
pca = PCA(n_components=2, random_state=42)
Z = pca.fit_transform(X_scaled)

print("explained_variance_ratio:",
      pca.explained_variance_ratio_)
```

PCA لا يصلح مصفوفة سيئة. قبل التطبيق يجب التأكد من عدم وجود NaN غير مقصودة، وأن الأعمدة العددية لها معنى ووحدات متوافقة، وأن القياس أو التوحيد تم وفق بروتوكول محدد. إذا كانت ميزة atomic_number موجودة مع atomic_weight بمقاييس مختلفة، قد تهيمن إحدى الميزات على التباين قبل التقييس. لذلك يجب أن يوضح التقرير هل استُخدم StandardScaler أم تحليل مباشر على القيم الخام، وما النسخة التي أنتجت المصفوفة.

PCA does not repair a bad matrix. Before applying it, check for unexpected NaNs, meaningful numeric columns, compatible units, and a defined scaling protocol. Atomic number and atomic weight have different scales, so one can dominate variance before scaling. The report should state whether StandardScaler was used or PCA was applied to raw values, and which feature-build version produced the matrix.

مذكرة مختبر: التحقق من المصفوفات قبل PCA

Lab Note: Validate Matrices Before PCA

PYTHON

```
ANGSTROM_TO_BOHR = 1.889726125

def angstrom_to_bohr(x):
    return x * ANGSTROM_TO_BOHR

def bohr_to_angstrom(x):
    return x / ANGSTROM_TO_BOHR

x = 0.735
round_trip = bohr_to_angstrom(
```

```
    angstrom_to_bohr(x)
)

assert abs(round_trip - x) < 1e-12
print(round_trip)
```

التحويل بين Angstrom وBohr أو بين eV وHartree يجب أن يكون وظيفة مستقلة يمكن اختبارها. الخطأ في عامل التحويل قد يغير الهندسة أو الطاقة ثم يظهر لاحقًا كخطأ كمومي لا علاقة له بالدائرة. لذلك نختبر التحويل العكسي ونقارن القيمة بعد التحويل والعودة ضمن حد عددي معلوم. كما يجب تخزين الوحدة الأصلية والوحدة المستهدفة في سجل التجربة.

Conversions such as Angstrom to Bohr or eV to Hartree should be independent, testable functions. A wrong conversion factor can change geometry or energy and later appear as a quantum error unrelated to the circuit. Test round trips and compare the returned value within a defined numerical tolerance. Store both original and target units in the experiment record.

مذكرة مختبر: اختبار الوحدات والتحويلات

Lab Note: Testing Units and Conversions

PYTHON

```
import hashlib
import json

params = {
    "basis": "sto3g",
    "active_electrons": 2,
    "active_orbitals": 2,
}

payload = json.dumps(
    params, sort_keys=True
).encode("utf-8")

print(hashlib.sha256(payload).hexdigest())
```

إصدار البرنامج وحده لا يكفي؛ النتيجة قد تتغير بسبب البيانات أو النموذج أو المعاملات. لذلك يمكن تعريف run_id يربط data_version وsoftware_version وmodel_version وparameters_hash. إذا أعيد تشغيل التجربة بنفس العقد يجب أن يكون الفرق بين النتيجتين قابلًا للتفسير. وإذا تغيرت إحدى هذه العناصر، تنشأ نسخة تشغيل جديدة بدل الكتابة فوق السجل السابق.

Software version alone is not enough; a result can change because of data, model, or parameters. Define a run_id linking data_version, software_version, model_version, and a parameter hash. If the experiment is rerun under the same contract, differences should be explainable. If one contract component changes, create a new run rather than overwriting the previous record.

مذكرة مختبر: إدارة الإصدارات العلمية

Lab Note: Scientific Version Management

PYTHON

```
from pathlib import Path
import hashlib

p = Path("data/elements.csv")
digest = hashlib.sha256(p.read_bytes()).hexdigest()

print("dataset:", p.name)
print("sha256:", digest)
```

لا يكفي أن نسمي الملف elements.csv؛ يجب أن تكون له نسخة معروفة وبصمة ووقت بناء. إذا تغير مصدر الوزن الذري أو التصنيف، فإن النموذج الذي تدرب على النسخة القديمة لا ينبغي أن يعاد تشغيله على الجديدة دون تسجيل التغيير. في سجل التجربة يكفي أحيانًا hash للملف مع اسم المصدر والإصدار، لكنه يصبح ضروريًا عندما تتعدد النسخ. هذا يربط البيانات بالنتيجة ويمنع أن يتحول ملف جديد إلى «نفس البيانات» في التقرير.

Naming a file elements.csv is not enough; it should have a known version, fingerprint, and build time. If an atomic-weight source or classification changes, a model trained on the old dataset should not silently be rerun on the new one. An experiment record can store a file hash together with source and version. This links data to results and prevents a new file from being described as “the same data” without evidence.

مذكرة مختبر: النسخة المرجعية للبيانات

Lab Note: The Reference Dataset Version

PYTHON

```
result = {
    "run_id": "run-014",
    "metric": "energy",
    "value": -1.1373,
    "unit": "Hartree",
    "data_version": "h2-sto3g-v1",
    "software_version": "qiskit-2.5.0",
}

required = {
    "run_id", "metric", "value",
    "unit", "data_version",
    "software_version",
}

assert required <= result.keys()
print("result context: PASS")
```

الرقم النهائي في تقرير علمي لا يكفي وحده. يجب أن يعرف القارئ نوع المقياس والوحدة ومصدر البيانات وإصدار البرنامج والـrun_id. هذه القاعدة تجمع SQL وOOP وAI والكم في فكرة واحدة: النتيجة ليست قيمة فقط، بل قيمة مع سياقها. إذا لم نستطع إعادة بناء السياق، فالرقم قد يكون مفيدًا كذاكرة لكنه لا يصلح كدليل قابل للمراجعة.

A final number in a scientific report is not sufficient by itself. The reader should know metric type, unit, data source, software version, and run_id. This rule connects SQL, OOP, AI, and quantum computing: a result is a value together with context. If the context cannot be reconstructed, the number may be a useful note but is not an auditable piece of evidence.

مذكرة مختبر: قاعدة «لا نتيجة بلا سياق»

Lab Note: No Result Without Context

SQL

```
ALTER TABLE experiments
ADD COLUMN backend_type TEXT;

ALTER TABLE experiments
ADD COLUMN backend_name TEXT;
```

```
ALTER TABLE experiments
ADD COLUMN shots INTEGER;
```

لا ينبغي أن يكتب البرنامج نتيجة المحاكي والجهاز الحقيقي في نفس السجل دون تمييز. يجب أن يحمل كل تشغيل backend_type مثل simulator أو hardware، مع backend_name وshots. هذا يمنع مقارنة غير واضحة بين نتيجة مثالية ونتيجة متأثرة بالضوضاء. كما يجعل التقرير قادرًا على تجميع النتائج حسب نوع التنفيذ.

The program should not store simulator and real-hardware results without distinguishing them. Each run should carry backend_type such as simulator or hardware, together with backend_name and shots. This prevents ambiguous comparisons between ideal and noisy results and allows reports to group results by execution type.

مذكرة مختبر: فصل المحاكي عن الجهاز الحقيقي

Lab Note: Separating Simulator and Hardware Runs

SQL

```
CREATE TABLE feature_provenance (
    feature_name TEXT PRIMARY KEY,
    source_field TEXT NOT NULL,
    transformation TEXT NOT NULL,
    unit TEXT,
    data_version TEXT NOT NULL
);
```

عندما تنتقل خاصية من SQL إلى نموذج تعلم آلي يجب أن نعرف من أين جاءت وكيف تحولت. يمكن أن يحمل سجل الميزة اسم الحقل الأصلي، والتحويل، والوحدة، والإصدار. هذا مفيد عند اكتشاف أن نموذجًا تغير لأن atomic_weight تحولت من مصدر إلى آخر. provenance ليست زينة في التقرير؛ هي جزء من القدرة على تفسير النتيجة.

When a property moves from SQL into a machine-learning model, its origin and transformations should be known. A feature record can store source field, transformation, unit, and version. This is useful when a model changes because atomic_weight came from a different source. Provenance is not decorative reporting; it is part of the ability to explain a result.

مذكرة مختبر: سجل مصدر الميزة

Lab Note: Feature Provenance Records

PYTHON

```
import numpy as np

distances = np.linspace(0.65, 0.85, 5)

for d in distances:
    print(
        "H2 bond distance (A):",
        float(d)
    )
    # Each distance becomes a separate
    # experiment record in the real workflow.
```

عند حساب طاقة جزيء يجب أن نعرف هل الهندسة ثابتة أم نبحث عن هندسة أقل طاقة. إذا تغيرت المسافة بين الذرات في كل تشغيل، فلا يمكن تفسير تغير الطاقة على أنه أثر للخوارزمية وحدها. يمكن إجراء sweep صغير للمسافة، حفظ كل نقطة كتشغيل مستقل، ثم رسم الطاقة مقابل المسافة. هذا الأسلوب يوضح للقارئ الفرق بين تحسين المعاملات داخل VQE وبين تغيير النموذج الفيزيائي نفسه.

When computing molecular energy, we must know whether geometry is fixed or optimized. If bond distance changes between runs, an energy difference cannot be attributed only to the algorithm. A small distance sweep can store each point as an independent run and plot energy versus distance. This shows the reader the difference between optimizing VQE parameters and changing the physical model itself.

مذكرة مختبر: اختبار التقارب الهندسي

Lab Note: Geometry-Convergence Testing

PYTHON

```
experiment = {
    "molecule": "CO2",
    "basis": "sto3g",
    "geometry": "O-C-O, 1.16 A bonds",
    "active_electrons": 2,
    "active_orbitals": 2,
}
print(experiment)
```

قاعدة الأساس ليست إعدادًا تجميليًا. تغييرها يغير التمثيل العددي للمدارات وبالتالي الهاملتوني والنتيجة. في مثال H_2O أو CO_2 يجب أن يسجل التقرير اسم basis set، والهندسة، وطريقة SCF، والفضاء النشط. إذا تغيرت قاعدة الأساس فلا تقارن النتيجة الجديدة بالقديمة كما لو أن التجربتين متطابقتان. يمكن أن يكون الهدف التعليمي هو توضيح هذا الفرق نفسه.

A basis set is not a cosmetic setting. Changing it changes the numerical representation of orbitals and therefore the Hamiltonian and result. For H_2O or CO_2, the report should record basis set, geometry, SCF method, and active space. If the basis changes, the new result should not be treated as identical to the old experiment. The difference itself can be the educational objective.

مذكرة مختبر: اختيار قاعدة الأساس في الكيمياء الكمومية

Lab Note: Basis-Set Choice in Quantum Chemistry

PYTHON

```
def classify_evidence(level):
    allowed = {"observational", "experimental", "review"}
    if level not in allowed:
        raise ValueError("unknown evidence level")
    return level

print(classify_evidence("observational"))
```

عند ربط العناصر بالبروتينات أو الأمراض الجينية يجب ألا يتحول وجود علاقة في قاعدة البيانات إلى ادعاء سببي. الارتباط قد ينتج من متغير مربك أو من اختيار العينة أو من اختلاف في القياس. لذلك يخزن Gene Resistance Tracker evidence_source وevidence_level، ويترك التفسير السريري لطبقة مستقلة. هذا التصميم يحمي النظام من تحويل سجل أدبي إلى حكم طبي.

When linking elements to proteins or genetic disease observations, a database relationship must not become a causal claim. Association can arise from confounding, sampling choices, or measurement differences. Gene Resistance Tracker therefore stores evidence_source and evidence_level while leaving clinical interpretation to a separate layer. This protects the system from turning a literature record into a medical judgment.

مذكرة مختبر: الفروق بين الارتباط والسببية

Lab Note: Association Is Not Causation

بوابة الإغلاق العلمي والتحريري

Scientific and Editorial Closure Gate

PYTHON

```python
checks = {
    "scientific_scope": True,
    "bilingual_pairing": True,
    "code_extractable": True,
    "images_intact": True,
    "periodic_table_complete": True,
    "searchable_pdf": True,
    "toc_after_layout": True,
}

assert all(checks.values())
print("FINAL PUBLICATION GATE: PASS")
```

المراجع والمصادر

References & Sources

المصادر العلمية الأساسية

Core Scientific Sources

1. IUPAC – الاتحاد الدولي للكيمياء البحتة والتطبيقية

https://iupac.org/

2. NIST – المعهد الوطني للمعايير والتقنية

https://www.nist.gov/

3. PubMed – قاعدة البيانات الطبية الحيوية

https://pubmed.ncbi.nlm.nih.gov/

4. NCBI – المركز الوطني لمعلومات التقانة الحيوية

https://www.ncbi.nlm.nih.gov/

5. Royal Society of Chemistry – الجمعية الملكية للكيمياء

https://www.rsc.org/

6. Los Alamos National Laboratory – مختبر لوس ألاموس الوطني (بيانات العناصر)

https://periodic.lanl.gov/

المراجع البرمجية والتقنية

1. Python Software Foundation – وثائق Python الرسمية

https://docs.python.org/

2. NumPy Documentation – وثائق NumPy

https://numpy.org/doc/

3. Pandas Documentation – وثائق Pandas

https://pandas.pydata.org/docs/

4. Qiskit Documentation – وثائق Qiskit

https://docs.quantum.ibm.com/

5. Scikit-learn Documentation – وثائق Scikit-learn

https://scikit-learn.org/stable/

6. SQLite Documentation – وثائق SQLite

https://sqlite.org/docs.html

scikit-learn Documentation — Probability calibration and evaluation metrics.

Dask Documentation — DataFrame best practices, partitioning, and Parquet workflows.

IBM Quantum Runtime Documentation — EstimatorV2 and SamplerV2; official API and migration guidance.

المراجع الأكاديمية

Academic References

1. Peruzzo, A., et al. (2014). A variational eigenvalue solver on a photonic quantum processor. Nature Communications, 5, 4213.

2. McClean, J. R., et al. (2016). The theory of variational hybrid quantum-classical algorithms. New Journal of Physics, 18, 023023.

3. Kandala, A., et al. (2017). Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. Nature, 549, 242-246.

4. Cao, Y., et al. (2019). Quantum chemistry in the age of quantum computing. Chemical Reviews, 119, 10856-10915.

5. Tilly, J., et al. (2022). The Variational Quantum Eigensolver: A review of methods and best practices. Physics Reports, 986, 1-128.

6. Harrow, A. W., Hassidim, A., & Lloyd, S. (2009). Quantum algorithm for linear systems of equations. Physical Review Letters, 103, 150502.

7. Sapova, M. D., & Fedorov, A. K. (2022). Variational quantum eigensolver techniques for simulating carbon monoxide oxidation. Communications Physics, 5, 199.

8. IBM Quantum Documentation. Qiskit Runtime and V2 primitives documentation.

9. Amazon Web Services. Amazon Braket developer documentation for QPUs and simulators.

10. Dask Documentation. Distributed and larger-than-memory data processing.

11. PyTorch Documentation. Tensor computation and deep-learning workflows.

12. TensorFlow Documentation. Keras and deep-learning workflows.

المراجع التربوية

Educational References

1. Enevoldsen, K. (2010). The Periodic Table of the Elements, in Pictures. elements.wlonk.com.

2. Gray, T. (2009). The Elements: A Visual Exploration of Every Known Atom in the Universe. Black Dog & Leventhal.

3. Scerri, E. (2019). The Periodic Table: Its Story and Its Significance. Oxford University Press.

الحوكمة هنا طبقة تنفيذية: طلب، موافقة، تشغيل، سجل تدقيق، ثم إصدار. يمكن أن يحتوي approval على purpose وdata_scope وmodel_version وrisk_class وreviewer وexpires_at. قبل التشغيل يتحقق النظام من أن القرار صالح وأن النطاق متوافق. بعد التشغيل يرتبط run_id بالمدخلات والمخرجات والنسخ.

Governance here is an executable layer: request, approval, run, audit trail, and release. An approval can contain purpose, data_scope, model_version, risk_class, reviewer, and expires_at. Before execution, the system checks that the decision is valid and the scope matches. After execution, run_id links inputs, outputs, and versions.

بطاقة المرجع: بطاقات مرجعية تشغيلية

Reference Card: Operational Reference Cards

بطاقات مرجعية تشغيلية — الحوكمة والإصدار

Operational Reference Cards — Governance and Release

يجب تحويل الدائرة إلى backend. التنفيذ على جهاز حقيقي ليس مجرد استبدال اسم المحاكي باسم backend صيغة متوافقة مع الهدف، ثم تسجيل shots والنسخ البرمجية وعدد ومعرف المهمة. وتدعم جلسات وتنفيذًا، SamplerV2 وEstimatorV2 الحديثة تستخدم Runtime واجهات مخصصًا. يجب اختبار البرنامج محليًا قبل استهلاك موارد الجهاز.

مع بيئة أسرار مناسبة، QiskitRuntimeService لا تحفظ رمز الدخول داخل المشروع. استخدم وسجل النتيجة الوصفية فقط. إذا فشلت المهمة، احفظ الحالة والسبب والمعرف. وإذا استخدمت تخفيف الخطأ، فاذكر التقنية والخيارات حتى يمكن تفسير الفرق بين القراءة الخام والنتيجة المعالجة.

Real-hardware execution is not just replacing a simulator name with a backend. The circuit must be made compatible with the target, and

the run should record backend, software versions, shot count, and job identifier. Modern Runtime interfaces use EstimatorV2 and SamplerV2 and support session-based execution. Test the program locally before consuming hardware resources.

Never store credentials inside the project. Use QiskitRuntimeService with an appropriate secret-management mechanism and record metadata rather than tokens. If a job fails, store its status, reason, and identifier. If error mitigation is used, document the method and settings so the difference between raw and processed results remains interpretable.

بطاقة المرجع: بطاقات مرجعية تشغيلية

Reference Card: Operational Reference Cards

بطاقات مرجعية تشغيلية — العتاد وRuntime

Operational Reference Cards — Hardware and Runtime

ابدأ دائمًا من السؤال الفيزيائي. اختيار الهندسة وقاعدة الأساس والفضاء النشط يحدد Hamiltonian. في H_2 يمكن بناء مثال صغير ومقارنته بـ Hartree–Fock الذي سيصل إلى الدائرة الكمومية. وFCI. عند الانتقال إلى H_2O أو CO_2، يجب ألا يوحي عدد 4–6 كيوبت بأنه تم تمثيل كل الإلكترونات والمدارات؛ إنه مثال على مسار mapping أو نموذج مختزل.

يحل HHL أنظمة خطية بشروط محددة، وليس حلًا مباشرًا عامًا لمعادلة شرودنجر. مسائل القيم الذاتية للطاقة لها خوارزميات مثل VQE أو تقدير الطور. وفي كل حالة يجب تسجيل خط الأساس، وحجم الخطأ، وإعدادات المحاكي أو الجهاز، وعدد shots، وطريقة معالجة الضوضاء.

Always begin from the physical question. Geometry, basis set, and active-space choices define the Hamiltonian that reaches the quantum circuit. For H_2, a small example can be compared with Hartree–Fock and FCI. When moving to H_2O or CO_2, a 4–6-qubit circuit must not be presented as a complete electronic representation; it is an example of mapping or a reduced model.

HHL solves linear systems under specific assumptions; it is not a general direct solver for the Schrödinger equation. Energy eigenvalue problems can use algorithms such as VQE or phase estimation. In every case, record the baseline, error magnitude, simulator or hardware settings, shot count, and noise-treatment method.

بطاقة المرجع: بطاقات مرجعية تشغيلية

Reference Card: Operational Reference Cards

بطاقات مرجعية تشغيلية — الكيمياء والحوسبة الكمومية

Operational Reference Cards — Chemistry and Quantum Computing

يجب فحص التكرار، VAE أو GAN النموذج التوليدي يولد مرشحين، ولا يمنحهم صفة الحقيقة. مع يجب LLM والصلاحية، ومدى الانحراف عن بيانات التدريب، ثم حساب الخواص بطريقة مستقلة. ومع والمصادر المرجعية هي طبقة الحقيقة، بينما يتولى النموذج صياغة السؤال أو تلخيص SQL أن تكون الأدلة.

في نظام توصية متعدد الأهداف، خزّن الأهداف والأوزان والقيود مع كل تشغيل. المرشح الذي يفوز

تحت أوزان معينة ليس فائزًا مطلقًا. وعندما تكون البيانات قليلة، يجب أن يذكر التقرير أن الناتج استكشافي وأن الحاجة إلى تحقق مستقل لم تختف.

A generative model produces candidates; it does not make them true. With GANs or VAEs, check duplicates, validity, distribution shift, and then calculate properties independently. With an LLM, SQL and documented sources should remain the truth layer, while the model formulates queries or summarizes evidence.

For multi-objective recommendation, store objectives, weights, and constraints with each run. A candidate that wins under one weighting is not an absolute optimum. When data are limited, the report should explicitly label the output as exploratory and state that independent validation is still required.

بطاقة المرجع: بطاقات مرجعية تشغيلية

Reference Card: Operational Reference Cards

بطاقات مرجعية تشغيلية — الذكاء الاصطناعي التوليدي

Operational Reference Cards — Generative AI

التعلم الآلي في هذا المشروع أداة للاستكشاف والتنبؤ، وليس بديلًا عن القياس. يجب فصل التدريب عن الاختبار، وتوثيق seed، ومنع تسرب المعلومات بينهما. إذا كانت السجلات مرتبطة بالعنصر نفسه أو المصدر نفسه، قد يكون التقسيم العشوائي مضللًا؛ يمكن أن يكون GroupKFold أو تقسيمًا حسب المصدر أنسب.

لا تستخدم accuracy وحدها. للتصنيف يمكن مقارنة precision وbalanced accuracy

وrecall وF1، وللاحتمالات يمكن فحص المعايرة و Brier score وlog loss. إذا كانت النتيجة ستستخدم لترتيب مرشحين، يجب تسجيل threshold ودالة القرار وإصدار النموذج.

Machine learning in this project is a tool for exploration and prediction, not a replacement for measurement. Separate training and testing, record the seed, and prevent information leakage. If records are related to the same element or source, random splitting may be misleading; GroupKFold or source-based splitting can be more appropriate.

Do not rely on accuracy alone. For classification, compare balanced accuracy, precision, recall, and F1; for probabilistic outputs, evaluate calibration, Brier score, and log loss. If predictions rank candidates, record the threshold, decision rule, and model version.

بطاقة المرجع: بطاقات مرجعية تشغيلية

Reference Card: Operational Reference Cards

بطاقات مرجعية تشغيلية — التعلم الآلي

Operational Reference Cards — Machine Learning

لا يبدأ التوسع باختيار Spark أو Dask؛ يبدأ بسؤال: هل البيانات أكبر من الذاكرة، وهل يمكن تقسيم العمل؟ إذا كانت 118 سجلًا أو ملفًا صغيرًا، فإن SQLite وPandas أبسط. عندما تكبر البيانات، يصبح حجم partition، وطريقة التقسيم، وعدد المهام، وكلفة shuffle عوامل حقيقية في الأداء.

Parquet مناسب للبيانات الجدولية الكبيرة لأنه يحتفظ بالمخطط ويدعم القراءة العمودية. لكن partitioning سيئ قد يخلق عددًا هائلًا من الملفات الصغيرة أو يفرض shuffle مكلفًا. سجّل

وإصدار المخطط وحجم الملفات، واختبر القراءة على عينة مماثلة قبل الانتقال إلى partition key cluster.

Scaling does not begin by choosing Dask or Spark; it begins by asking whether the data exceed memory and whether the work can be partitioned. For 118 records or a small file set, pandas and SQLite are simpler. As data grow, partition size, task count, shuffle cost, and partitioning strategy become real performance factors.

Parquet is well suited to large tabular datasets because it carries schema information and supports column-oriented access. Poor partitioning can nevertheless create huge numbers of small files or expensive shuffles. Record the partition key, schema version, and file-size strategy, and benchmark a representative sample before moving to a cluster.

بطاقة المرجع: بطاقات مرجعية تشغيلية

Reference Card: Operational Reference Cards

بطاقات مرجعية تشغيلية — Parquetو Dask

Operational Reference Cards — Dask and Parquet

كل قيمة عددية في المشروع يجب أن تحمل سياقها: الوحدة، المصدر، الإصدار، وربما عدم اليقين. القيمة المفقودة ليست صفرًا، والقيمة المشتقة ليست قياسًا. عند بناء المصفوفة، احتفظ بهوية الصف والعمود وإصدار مجموعة البيانات. إذا غيّرت طريقة معالجة القيم المفقودة أو التقييس، أنشئ إصدارًا جديدًا للميزة بدل الكتابة فوق النتيجة القديمة.

اختبارات الجودة يجب أن تعمل قبل التحليل: عدد السجلات، التفرد، نطاق القيم، الوحدات، القيم المفقودة،

واتساق العلاقات. يمكن أن يفشل الاختبار عمدًا إذا وجد قيمة غير منطقية؛ فالفشل المبكر أفضل من إنتاج نموذج يبدو صحيحًا على بيانات فاسدة.

Every numerical value in the project needs context: unit, source, version, and, when available, uncertainty. A missing value is not zero, and a derived value is not a measurement. When building a matrix, preserve row and column identity and the dataset version. If missing-value handling or scaling changes, create a new feature version rather than overwriting the old result.

Quality checks should run before analysis: record count, uniqueness, value ranges, units, missingness, and relationship consistency. A test should be allowed to fail deliberately when a value violates a rule; early failure is preferable to producing a model that looks correct on corrupted data.

بطاقة المرجع: بطاقات مرجعية تشغيلية

Reference Card: Operational Reference Cards

بطاقات مرجعية تشغيلية — جودة البيانات

Operational Reference Cards — Data Quality

للمشروع، ابدأ بالهوية قبل الخاصية. العدد الذري هو مفتاح العنصر، والرمز SQL عند بناء مخطط يجب أن يكون فريدًا، والعلاقات مع المجموعة والدورة والفئة يجب أن تكون قابلة للتحقق. استخدم NOT NULL عندما تكون هناك مجموعة CHECKللحقول التي لا يمكن أن يعمل السجل بدونها، و FOREIGN KEYقيم محددة، و عندما تعتمد القيمة على كيان آخر. لا تستخدم النص الحر لتخزين علاقة يمكن تمثيلها بمفتاح.

المعاملة تحدد حدود التغيير. إذا كانت عملية التحميل تتكون من خطوات متعددة، يجب أن يكون من الممكن معرفة هل نجحت كلها أم فشلت كلها. وفي العمليات القابلة لإعادة المحاولة، تمنع المفاتيح والقيود الفريدة تكرار السجل. أما تحديث المخطط فيحتاج إصدارًا ومسارًا للرجوع واختبارًا على نسخة من البيانات. بهذا يصبح SQL جزءًا من الحوكمة التقنية، لا مجرد طبقة تخزين.

When designing the project SQL schema, define identity before properties. Atomic number is the element identity, symbol should be unique, and relationships to group, period, and category should be enforceable. Use NOT NULL for required fields, CHECK for constrained value sets, and FOREIGN KEY when a value depends on another entity. Do not store a relationship as free text when it can be represented by a key.

Transactions define change boundaries. If loading consists of multiple steps, the system should make it clear whether the whole operation succeeded or failed. For retryable operations, keys and uniqueness constraints prevent duplicate records. Schema changes need a version, a rollback path, and testing on a copy of the data. SQL therefore becomes part of technical governance, not merely storage.

بطاقة المرجع: بطاقات مرجعية تشغيلية

Reference Card: Operational Reference Cards

SQL — بطاقات مرجعية تشغيلية

Operational Reference Cards — SQL

```
try:
    result = run_stage()
    status = "success"
```

```
except Exception as exc:
    result = None
    status = "failed"
    error_message = str(exc)
```

PYTHON

الفشل لا يتحول إلى قيمة افتراضية؛ يسجل كحالة يمكن تحليلها.

Failure should not become a default value; it is recorded as an analyzable state.

8. تسجيل فشل المرحلة

8. Record stage failure

```
from datetime import datetime, timezone
from uuid import uuid4

run_id = str(uuid4())
created_at = datetime.now(timezone.utc).isoformat()
print(run_id, created_at)
```

PYTHON

سجل التشغيل يجب أن يكون قابلًا للربط بالمدخلات والمخرجات.

A run record should be linkable to inputs and outputs.

7. تسجيل تشغيل

7. Record a run

```
from pyscf import gto, scf, fci

mol = gto.M(atom="H 0 0 0; H 0 0 0.74", basis="sto-3g")
mf = scf.RHF(mol).run()
e_fci, _ = fci.FCI(mol, mf.mo_coeff).kernel()
```

PYTHON

المرجع الكلاسيكي يحدد إن كانت نتيجة VQE ضمن النطاق المتوقع.

The classical reference indicates whether a VQE result is in the expected range.

6. Hartree–Fock and FCI reference

6. مرجع Hartree–Fock وFCI

```
import dask.dataframe as dd

df = dd.read_parquet("data/large_features/")
result = df.groupby("category")["value"].mean().compute()
```

PYTHON

هذا المثال يوضح نمطًا مناسبًا للبيانات الكبيرة، وليس للجدول الدوري الصغير.

This pattern is for large data, not for the intentionally small periodic-table dataset.

5. Dask with Parquet

5. Dask مع Parquet

```
import numpy as np

assert np.isfinite(X_scaled).all()
assert X.shape[0] == len(identifiers)
```

PYTHON

اختبار بسيط قبل النموذج يمنع انتقال خطأ واضح إلى مرحلة التعلم.

A simple pre-model check prevents obvious errors from entering the learning stage.

4. فحص عدم وجود قيم غير منطقية

4. Reject invalid values

```
from sklearn.decomposition import PCA

pca = PCA(n_components=2, random_state=42)
X_pca = pca.fit_transform(X_scaled)
print(pca.explained_variance_ratio_)
```

PYTHON

يجب حفظ عدد المكونات وإصدار البيانات حتى يمكن إعادة الرسم أو المقارنة.

Store component count and dataset version so the result can be reproduced or compared.

3. PCA مع تسجيل الإصدار

3. PCA with recorded configuration

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

PYTHON

التقييس يسبق PCA أو بعض النماذج عندما تختلف مقاييس الخصائص بشدة.

Scaling should precede PCA or some models when feature scales differ strongly.

2. تقييس الخصائص

2. Scale numerical features

```
import numpy as np

identifiers = np.array([
    (1, "H"),
    (2, "He"),
], dtype=[("atomic_number", "i4"), ("symbol", "U2")])
X = np.array([[1.008, 0.09], [4.003, 0.18]], dtype=float)
```

PYTHON

عند إنشاء المصفوفة احتفظ بمعرفات العناصر منفصلة عن القيم العددية.

When creating a matrix, preserve element identifiers separately from numerical values.

1. استخراج هوية الصفوف

1. Preserve row identity

دليل Python التحليلي — أمثلة قابلة للاختبار

Python Analysis Guide — Testable Examples

```
SELECT backend, software_version, energy, run_id
FROM quantum_results
WHERE system_id = :system_id
ORDER BY software_version, run_id;
```

SQL

يساعد هذا الاستعلام على كشف تغير النتائج بعد تحديث البيئة.

This query helps detect result changes after an environment update.

10. مقارنة نتائج كمومية بإصداراتها

10. Compare quantum results by software version

```
SELECT g.gene_id, v.variant_id, e.source_reference, e.confidence
FROM genes AS g
JOIN variants AS v ON v.gene_id = g.gene_id
JOIN evidence AS e ON e.variant_id = v.variant_id
WHERE g.gene_id = :gene_id;
```

SQL

الربط هنا ينظم الأدلة ولا يحولها إلى حكم سريري.

This join organizes evidence without turning it into a clinical judgment.

9. ربط دليل مقاومة بجين ومتغير

9. Join resistance evidence to gene and variant

```
SELECT approval_id
FROM preapproval
WHERE decision = 'approved'
  AND expires_at >= CURRENT_TIMESTAMP
  AND data_scope = :data_scope
  AND model_version = :model_version;
```

SQL

قبل التوليد أو المحاكاة، يجب أن تكون الموافقة صالحة وغير منتهية ومطابقة للنطاق.

Before generation or simulation, approval must be valid, unexpired, and scope-compatible.

8. بوابة الموافقة المسبقة

8. Check a pre-approval gate

```
SELECT run_id, model_version, dataset_version, created_at
FROM analysis_runs
WHERE stage = 'ml'
ORDER BY created_at DESC
LIMIT 1;
```

SQL

يربط هذا الاستعلام نتيجة النموذج بإصدار الكود والبيانات، وهو شرط لإعادة الإنتاج.

This links a model result to code and data versions, which is required for reproducibility.

7. آخر تشغيل لنموذج

7. Find the latest model run

```
SELECT symbol, atomic_weight, source_reference
FROM elements
WHERE symbol = :symbol;
```

SQL

عندما يكون source_reference جزءًا من السجل، يمكن ربط النتيجة بالمرجع بدل فقدان سياقها.

When source_reference is stored with the record, a result can be linked back to its evidence.

6. تتبع مصدر قيمة

6. Trace a value to its source

```
SELECT symbol, COUNT(*) AS n
FROM elements
GROUP BY symbol
HAVING COUNT(*) > 1;
```

SQL

قبل قبول البيانات CI هذا اختبار سلامة يمكن تشغيله في.

This is an integrity check that can run in CI before data are accepted.

5. فحص تكرار الرموز

5. Check duplicate symbols

```
SELECT atomic_number, symbol
FROM elements
WHERE atomic_weight IS NULL
   OR density IS NULL;
```

SQL

القيمة المفقودة يجب أن تظهر صراحة قبل بناء المصفوفة، لا أن تتحول تلقائيًا إلى صفر.

Missing values should be exposed before matrix construction rather than silently converted to zero.

4. البحث عن القيم المفقودة

4. Find missing values

```
SELECT period, AVG(atomic_weight) AS mean_atomic_weight
FROM elements
GROUP BY period
ORDER BY period;
```

SQL

كيف تتحول العلاقة العلائقية إلى ملخص عددي يمكن مقارنته مع المخرجات GROUP BY يوضح البرمجية.

This shows how GROUP BY turns relational data into a numerical summary that can be compared with programmatic output.

3. متوسط الوزن الذري حسب الدورة

3. Average atomic weight by period

```
SELECT c.category_english, COUNT(*) AS element_count
FROM elements AS e
JOIN categories AS c ON c.category_id = e.category_id
GROUP BY c.category_english
ORDER BY element_count DESC;
```

SQL

يساعد هذا الاستعلام على فحص العلاقات الخارجية والتجميع دون تعديل البيانات.

This query checks foreign-key relationships and aggregation without changing data.

2. توزيع العناصر حسب الفئة

2. Count elements by category

```
SELECT atomic_number, symbol, name_english, atomic_weight
FROM elements
WHERE atomic_number = :atomic_number;
```

SQL

يختبر هذا الاستعلام أن الهوية الأساسية تعمل كما صممها المخطط.

This query tests that the primary identity behaves as designed.

1. العثور على عنصر بالعدد الذري

1. Find an element by atomic number

دليل الاستعلام والتنفيذ — أمثلة قابلة للاختبار

عقد الحقول التشغيلية — مرجع البيانات

Query and Execution Guide — Testable Examples

Operational Field Contract — Data Reference

هذا الجدول يحدد الحد الأدنى من عقود الحقول التي يحتاجها المسار من SQL إلى التحليل والتشغيل.

This table defines a minimum field contract for the path from SQL to analysis and execution.

الكيان / Entity	الحقل / Field	Type / النوع	القيد / Constraint	المعنى / Meaning
elements	atomic_number	INTEGER	PRIMARY KEY	الهوية العددية للعنصر / Element identity

الكيان / Entity	الحقل / Field	Type / النوع	القيد / Constraint	المعنى / Meaning
elements	symbol	TEXT	UNIQUE NOT NULL	الرمز الكيميائي / Chemical symbol
elements	name_arabic	TEXT	NOT NULL	الاسم العربي / Arabic name
elements	name_english	TEXT	NOT NULL	English / الاسم الإنجليزي name
elements	atomic_weight	REAL	CHECK / source	الوزن الذري وسياقه / Atomic weight and context
elements	group_number	INTEGER	FOREIGN KEY	المجموعة الدورية / Periodic group
elements	period	INTEGER	CHECK 1..7	الدورة / Period
elements	category_id	INTEGER	FOREIGN KEY	الفئة الكيميائية / Chemical category
elements	source_reference	TEXT	NOT NULL	مصدر القيمة / Value source
elements	measurement_unit	TEXT	NOT NULL when applicable	وحدة القياس / Measurement unit
elements	uncertainty	REAL	NULLABLE	عدم اليقين إن توفر / Uncertainty when available
analysis_runs	run_id	TEXT	PRIMARY KEY	هوية التشغيل / Run identity
analysis_runs	dataset_version	TEXT	NOT NULL	إصدار البيانات / Dataset version
analysis_runs	model_version	TEXT	NULLABLE	إصدار النموذج / Model version
analysis_runs	status	TEXT	CHECK	Run status / حالة التشغيل
analysis_runs	created_at	TEXT	NOT NULL	وقت التشغيل / Execution time
quantum_results	system_id	TEXT	NOT NULL	معرف النظام / System identifier
quantum_results	backend	TEXT	NOT NULL	المحاكي أو الجهاز / Simulator or backend
quantum_results	software_version	TEXT	NOT NULL	إصدار البرمجيات / Software version
quantum_results	shots	INTEGER	CHECK > 0	عدد القياسات / Number of shots
quantum_results	energy	REAL	NULLABLE	الطاقة المقدرة / Estimated energy
preapproval	approval_id	TEXT	PRIMARY KEY	هوية الموافقة / Approval identity
preapproval	data_scope	TEXT	NOT NULL	Data / نطاق البيانات scope
preapproval	expires_at	TEXT	NOT NULL	انتهاء الموافقة / Approval expiry

```
from __future__ import annotations
import hashlib
from pathlib import Path

def sha256_file(path: str) -> str:
    h = hashlib.sha256()
    with Path(path).open("rb") as f:
        for chunk in iter(lambda: f.read(1024 * 1024), b""):
            h.update(chunk)
    return h.hexdigest()

artifact = "reports/final_report.pdf"
print({"artifact": artifact, "sha256": sha256_file(artifact)})
```

PYTHON / artifact_hash.py

البصمة لا تثبت .run_id للملف الناتج وربطه بـ SHA-256 عند إنشاء تقرير أو مصفوفة، يمكن حفظ أن المحتوى صحيح علميًا، لكنها تثبت أن الملف الذي تمت مراجعته هو الملف الذي يحمل نفس البصمة. هذه خطوة عملية مهمة عندما تتعدد نسخ PDF وTXT والملفات التنفيذية.

When a report or matrix is produced, its SHA-256 hash can be stored with run_id. A hash does not prove scientific correctness, but it proves that the reviewed artifact is the artifact with the same fingerprint. This is useful when multiple PDF, TXT, and executable-file versions exist.

إثبات مصدر المخرج ببصمة ملف

Proving Output Provenance with a File Hash

```
from __future__ import annotations
import sqlite3
DB = "data/periodic_table.db"
with sqlite3.connect(DB) as con:
    rows = con.execute("""SELECT atomic_number, symbol, atomic_weight FROM elements ORDER BY
atomic_number""").fetchall()
assert len(rows) == 118
assert len({r[0] for r in rows}) == 118
assert all(isinstance(r[0], int) for r in rows)
assert all(r[1] for r in rows)
print("field contract: PASS")
```

PYTHON / validate_contract.py

لا يكفي أن يكون الحقل موجودًا في SQLite؛ يجب أن يمر بعقد واضح قبل أن يدخل NumPy أو التعلم الآلي. يختبر المثال التالي أنواعًا أساسية، ويمنع وجود العدد الذري المكرر، ويتحقق من أن كل صف يمكن ربطه بهوية عنصر. يمكن توسيع الاختبار ليشمل الوحدات والمصادر والقيم المفقودة.

A field being present in SQLite is not enough; it should pass an explicit contract before entering NumPy or machine learning. The following example checks basic types, prevents duplicate atomic numbers, and verifies that every row maps to an element identity.

The same pattern can be extended to units, provenance, and missingness.

فهرس المحتويات

Table of Contents

أرقام الصفحات مستخرجة من النسخة المرقمة قبل إدراج هذا الفهرس النهائي.

The page numbers were taken from the final paginated source before this TOC was appended.